ROME AIR DEVELOPMENT CENTER

AUTOMATED SNEAK CIRCUIT ANALYSIS TECHNIQUE

RADC

Systems Reliability & Engineering Division
Rome Air Development Center

AUTOMATED

SNEAK
CIRCUIT
ANALYSIS
TECHNIQUE

RADC

SCAT

June 1990

Systems Reliability & Engineering Division
Rome Air Develpment Center

TABLE OF CONTENTS PAGE

ABSTRACT

Standard Sneak Analysis procedures are costly from a time, money, and personnel perspective. The processing of design data available only during the latter portions of the development cycle are highly labor intensive and create difficulties in instituting a design change. Automated techniques for sneak analysis have been developed by several contractors for the workstation environment, but are either not completed or are proprietary. In fact, some of these techniques are only a subset of the sneak analysis procedure, namely design concerns. The Sneak Circuit Analysis Technique (SCAT) overcomes many of these deficiencies by providing a personal computer based system for real time identification (during the actual design phase) of sneak paths and design concerns. The tool utilizes an expert system shell to assist the analyst so that prior experience with sneak analysis is not necessary for performance. Both sneak circuits and design concerns are targeted by this tool with both digital and analog circuits being examined. SCAT focuses the analysis at the assembly level rather than the entire system so that most sneak problems can be identified and corrected by the responsible design engineer in a timely manner.

NOTE:

This report contains excerpts of the work done under RADC Contract F30602-87-C-0193 by J. Miller, SoHaR Incorporated. The RADC technical reports;

(1) RADC-TR-89-223, Sneak Circuit Analysis for the Common Man
OCT '89, SoHaR Incorporated,
(2) RADC-TR-90-109, Integration of Sneak Circuit Analysis with
Design - MAY '90, SoHaR Incorporated,

are the basis for this document. The demonstration SCAT program is supplied free of charge. The only request is that after using SCAT, that you write any comments, critiques, or suggestions for improvement on the enclosed comment sheet and send it to:

RADC/RBER
ATTN: Edward DePalma
Griffiss AFB NY 13441-5700

Thank you for your comments, they are appreciated.

INTRODUCTION

Failures are placed into two broad categories, equipment breakdown and human error. **Equipment breakdown** has a variety of causes, among them are thermal stress, vibrational stress, and wearout. There are tests and precautions that can be implemented to greatly reduce the occurrence of these failures. **Human error** is more ambiguous. Human engineering and extensive training are used to minimize human error. There is, however, one other type of failure that has the characteristics of both a mechanical and human failure, and that is a sneak circuit failure.

Sneak circuits are hidden paths within an electrical system. A failure occurs when one of these paths cause an unwanted function to take place or inhibit a desired function from being performed. The unique property of the sneak circuit failure is that the failure occurs when all components are operating within their specifications. The flaw causing the sneak is present within the design itself. So a sneak failure is a human error that manifests itself within a mechanism.

Since sneak circuits are hidden in the design, common failure prevention methods, such as stress analysis, derating, redundancy, or environmental screening will have little effect on a sneak circuit. The only preventive measure used today is an in-depth circuit analysis performed by a limited number of outside contractors who specialize in sneak circuit analysis. Performing a sneak circuit analysis using outside expertise is costly in time and money. These contractors require a high level design and many days to completely analyze it. However, if the sneaks are identified early in the design phase, then the corrections are made before the formation of the high level design thus the time and money that would have been spent on an extended analysis is saved. This is the reasoning behind the SCAT package developed by the Rome Air Development Center and SoHaR Incorporated.

SCAT is a PC-based software package that allows the designer to perform a detailed sneak circuit analysis as the design proceeds. The SCAT program uses the M.1 expert system as a logic shell and the OrCAD/SDT III schematic capture program to generate a netlist (OrCAD/SDT III and M.1 are the only purchases required). The SCAT package will analyze analog as well as digital circuitry. Sectional analysis can be performed on extended circuit networks. A detailed description of the operation of the SCAT package is given in Appendix A, SCAT Users Manual. SCAT is available through RADC, whereas, OrCAD/SDT III and M.1 are licensed materials which can be obtained from:

OrCAD/SDT III
OrCAD Systems Corporation
3175 N.W. Aloclek Drive
Hillsboro OR 97124
(503) 690-9881

M.1 Expert System
Teknowledge Inc.
1850 Embarcadero Rd.
P O BOX 10119
Palo Alto CA 94303
(415) 424-0500

For anyone who is unfamiliar with sneak circuits, Appendix B contains examples of actual sneak failures. Also, Appendix B lists common circuit designs which are susceptible to sneak failures and offers a description of the manual sneak circuit process. This was included to supply the user with background knowledge on sneak circuits.

As a summary, the following hardware and software needs are necessary for the operation of the SCAT program:

-IBM PC or compatible PC
-640Kb of memory
-10Mb hard drive
-M.1 Program disk
-OrCAD/SDT III netlist data
-SCAT demonstration disk

AUTOMATED SCA

This chapter describes the development and operation of a simplified, automated SCA procedure developed during this study. The intent of developing this procedure was to demonstrate the concept and feasibility of integrating an SCA tool with an automated design tool to provide a simple yet effective sneak analysis procedure. To this end, the procedure and the automated SCAT supporting it were constrained to a specific input domain (i.e., a netlist comprising circuit device types from a specified parts library and formatted in a specified manner) and to a subset of sneak clues (i.e., those associated with commonly encountered sneak conditions). However, the procedure can readily be extended to apply to a wider variety of input data and check for a larger number of conditions.

1 Overview

SCAT is a microcomputer based expert system for automatically identifying sneak paths and design concerns by processing circuit netlists generated by a CAD schematic capture tool. SCAT differs from conventional SCA techniques in that the latter are based upon the generation (usually automated) and analysis (mostly manual) of network trees to identify sneak paths. The proposed tool does not require network trees; in fact, it is particularly applicable to early phases of a design when detailed circuit data required for generating trees are not available.

The automated procedure provides the design engineer or reliability analyst with a simple tool for rapidly identifying and correcting sneak circuits and relevant design concerns. Identification of topological patterns is not required. Sneak paths are automatically identified for power switching circuitry. Design concerns relevant to sneak circuits are identified for analog or digital circuits. The procedure focuses the analysis on portions of the circuitry for which the analyst has design responsibility (or detailed understanding of its operation), e.g., a circuit card assembly or a subsystem such as power distribution. A more extensive analysis would require application of a conventional SCA. However, even in this case, prior use of the proposed procedure would minimize the number of remaining sneaks and thereby greatly reduce the cost impact and other concerns associated with correcting problems late in the design phase.

The automated procedure is based in part on the fact that sneak paths involve circuit branches that can conduct current in either direction depending upon the switching state of the circuit. SCAT searches for these bidirectional branches rather than perform the more complex task of searching for specific topological circuit patterns as done by conventional automated SCA techniques. The analyst's task is also reduced to evaluating the significance of specific sneak paths rather than applying "clue lists" to circuit patterns for identifying the sneak paths.

A significant issue that arises in regard to focusing the analysis at any one time to a portion of the system is the assurance that sneak paths associated with assembly or subsystem interfaces are not overlooked. This issue is addressed in two ways. First, the system compels the user to identify each interface port of a switching circuit in terms of it being a power input, ground return, or signal I/O. Interfaces to power and ground are labeled as such regardless of whether they respectively go to power and ground directly or through switched or unswitched loads, and they are included within the sneak path search. Second, the SCAT is intended to identify many (but not necessarily all) sneak conditions early in the design when interfaces may not yet be completely defined. In this manner, problems can be corrected early at minimal cost so that at a later development phase a more conventional SCA can be performed on the entire system and uncover the few, if any, remaining problems.

2 Description

The automated procedure comprises four major tasks:

1. Schematic capture/netlist generation.
2. Sneak path analysis.
3. Functionally oriented design concern analysis.
4. Device oriented design concern analysis.

These tasks have been computerized utilizing a concurrent engineering environment comprising a commercial schematic capture product and the expert system based SCAT. Schematic capture and netlist generation are performed by OrCAD/SDT III version 3.21 or later release. It is available through OrCAD Systems Corp., Hillsboro, Oregon.

The sneak path and design concern analyses are performed by the SCAT expert system, developed from an M.1 (Teknowledge, Inc.) expert system shell. The shell consists of the M.1 inference engine and facilities for developing and maintaining the SCAT knowledge base. M.1 was selected from among eleven, commercially available, MS-DOS based shells evaluated for the SCAT application. M.1 was selected because of its execution speed (the shell is coded in C rather than LISP which is used by many others), its rich repertory of syntactical functions (e.g., pattern-matched variables, a facility for rule looping, and a large number of built-in meta-facts and meta-propositions), and its open knowledge base (the rules are formatted in ASCII text).

The SCAT knowledge base consists of approximately 265 rules stored in nine data files. The knowledge base is augmented by approximately 650 lines of C source code implementing those portions of the analysis requiring intensive processing. These portions include reading the EDIF netlist and performing the sneak path search. The code was developed on Microsoft's QuickC environment and C 5.0 Optimizing Compiler. Information needed for maintaining the program, including extensively documented source code and knowledge base files, appears in Volume II of Integration of Sneak Analysis with Design. To maintain an audit trail, each file is prefaced by a header identifying the date of the last revision, its originator, and a brief description of it.

A functional diagram of the system appears in Figure 1. Prior to running SCAT, the user must generate a netlist of the circuit to be analyzed using the OrCAD schematic capture program. The netlist must be saved either on hard disk or floppy disk. SCAT reads and processes the netlist, as directed by the user, to identify sneak paths and design concerns. Utilizing the user friendly, consultation type interface provided by the SCAT expert system, the user must specify the name of the netlist file and the type of analysis to be performed (sneak path or design concern). For the sneak path analysis, the user must confirm that suspicious paths identified by SCAT are in fact sneak circuits (i.e., they inhibit desired functions or cause undesired outputs). The expert system provides assistance for this task. For the design concern analysis, the user must respond to prompts regarding technical details of the circuit under analysis. Assistance for this is available in the form of "help" messages. Explanations and possible solutions for identified design concerns are also available. Operational details, program limitations, and an example of the procedure are provided by the SCAT User's Manual in Appendix A. A general description of the major tasks involved in the procedure is presented next.

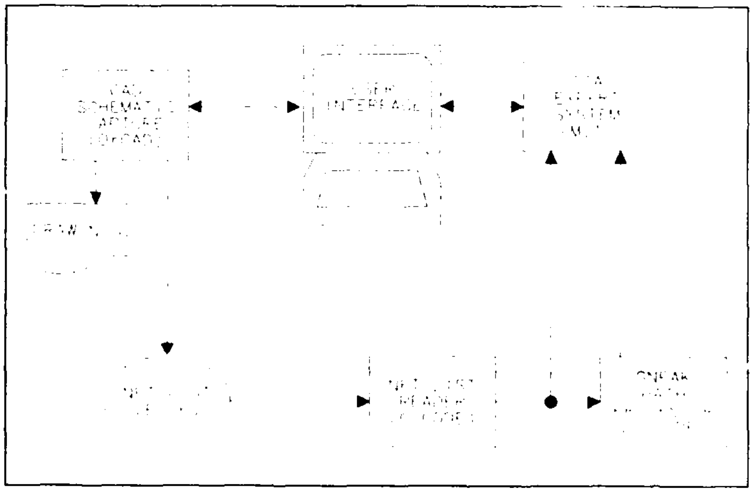


Figure 1. Computer Aided System for Sneak Analysis

2.1 Schematic Capture/Netlist Generation

The automated procedure requires that the circuit under analysis be captured and formatted by OrCAD/SDT III. A schematic is captured by drawing it on the screen using the various graphics and text editing features and the device symbol libraries provided by the program. All interfaces involving power and ground, whether direct from external sources or from external switches/drivers must be labeled using the OrCAD "module port" function (an option provided by the program for designating signal terminations). This will enable SCAT to account for all significant interfaces to portions of the system not under analysis. In addition, the terminals of all internal power sources (e.g., on-board batteries) must be similarly labeled to address potential sneaks involving power-to-power ties. After capturing the schematic, the netlist is generated and saved using the OrCAD "FlatEDIF" utility. When invoked by the user, this utility translates the captured schematic into an ASCII text file conforming to the Electronic Industries Association (EIA) Interim Standard No. 44 for EDIF version 1 1 0[1].

2.2 Sneak Path Analysis

After generating the netlist, the user enters SCAT and specifies the name of the netlist file to be processed. The user is then given the option of performing a sneak path analysis or the design concern analyses. The following discussion assumes the former has been selected.

Sneak path analysis is performed on power switching circuitry, i.e., circuits involving combinations of current interruption devices such as switches, relays, fuses, connectors, and transistors. During the analysis, all possible non-cyclic (i.e., non-intersecting), directed paths are automatically identified between every pair of power and power return points in the circuit (herein after referred to as the source node and sink node) specified by the user. To facilitate this path search, SCAT automatically models the following types of devices:

- Switch and relay contact arrangements (single and multiple pole/throw, break-before-make and make-before-break).
- Transistors (both bipolar and MOS) and diodes.
- Capacitors (under conditions of both AC and DC current flow).
- Other two-terminal passive devices (resistors, inductors, etc.).
- Multi-terminal passive devices (transformers, potentiometers, etc.).

The user may override (either globally or for specific devices) the default model assumed for switch and relay contact timing (break-before-make) and for capacitors (open circuit). Connections to integrated circuits are modeled as open circuits since paths are not traced through these devices.

[1] The more recent version, 2 0 0, was not available in time for this effort.

Following the path search, path sets are identified in which a common branch conducts current in both directions. These bidirectional branches are usually indicative of an undesired reverse current path, the distinguishing feature of a sneak circuit. Each reverse current path is displayed as a list of schematic reference designators of the devices that appear along the path, listed in the order of their appearance between the source node and sink node. Once identified, the user must confirm the validity of the path by considering operational constraints that may preclude certain switching states assumed by the path in question. In addition, their potential impact on system operation must be evaluated. The SCAT expert system provides guidance for these evaluations to the less experienced user. The guidance is in form of prompts regarding the location of critical loads and the timing of switches affecting the path.

An example of a reverse current path identified by SCAT is shown by the schematic in Figure 2. The circuit is a simplified version of the infamous Redstone missile/blockhouse interface that caused premature engine cutoff a few seconds after launch. The cause was determined to be the sneak path, highlighted on the schematic, between the launch command and engine cutoff relays that occurred when the ground umbilical separated a fraction of a second before the separate power umbilical. The netlist for this schematic was processed by SCAT, and the resulting screen corresponding to the reverse current path is shown in Figure 3. The path is identified by the part reference designators appearing in the schematic.

2.3 Functionally Oriented Design Concern Analysis

Functionally oriented design concerns address the following types of sneak conditions:

- power-to-power ties.
- inadvertent load power cutoff by logically AND'd switching devices.
- inadvertent load power enabling by logically OR'd switching devices.
- improper timing for power enabling and power cutoff.
- misleading indications and labels.

More than serving as clues for the analyst, these concerns compose a knowledge base of rules that are evaluated by SCAT with respect to the specific circuit being analyzed. In this manner, non-relevant clues are filtered, and wherever possible the user is directed to specific areas of the circuit under concern. As an example, consider the power-to-power tie highlighted in the schematic of Figure 4. The circuit is a portion of the FB111(A)B Pivot Pylon Weapon Station Circuitry that underwent a conventional SCA circa 1975 [BOEI75]. The power tie was identified by SCAT as shown by the screen in Figure 5. As shown in Figure 6, the user is also offered assistance by way of an explanation message and a possible solution. These aids are directed toward less experienced users.

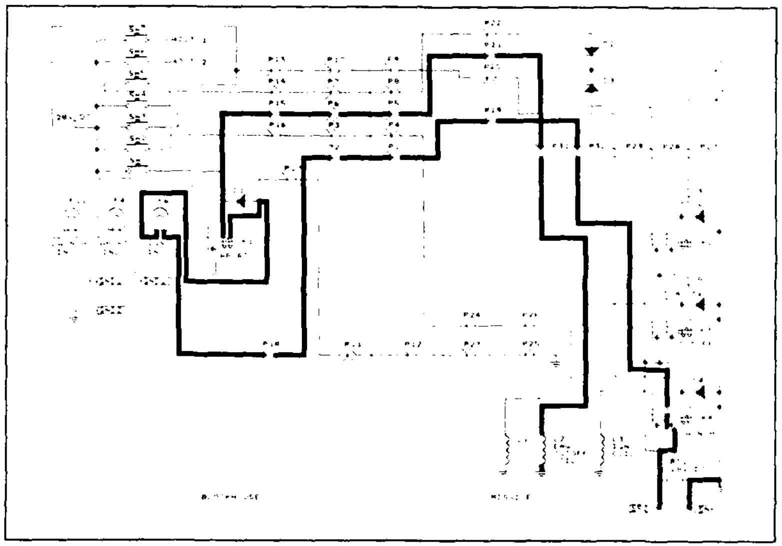
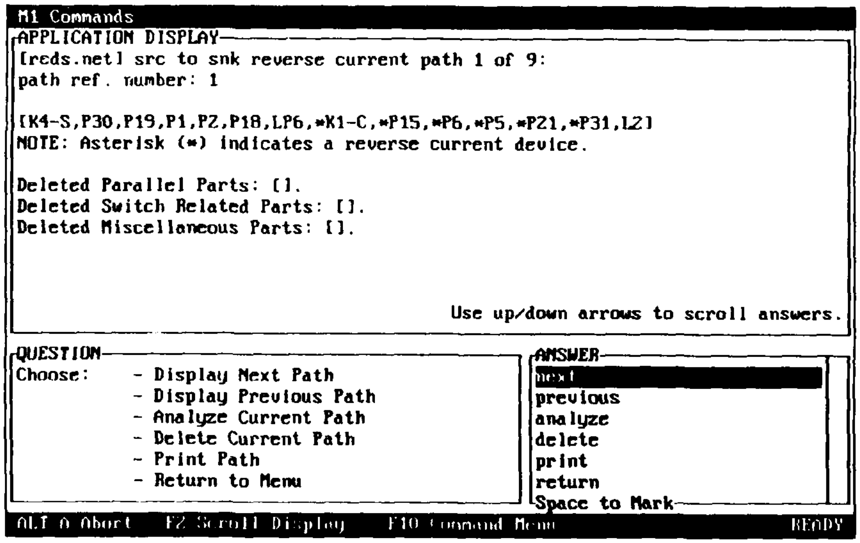


Figure 2. Reverse Current Path in Missile Launch Circuit



```
[reds.net] src to snk reverse current path 1 of 9:
path ref. number: 1

[K4-S,P30,P19,P1,P2,P18,LP6,*K1-C,*P15,*P6,*P5,*P21,*P31,L2]
NOTE: Asterisk (*) indicates a reverse current device.

Deleted Parallel Parts: [].
Deleted Switch Related Parts: [].
Deleted Miscellaneous Parts: [].

                                        Use up/down arrows to scroll answers.

Choose:       - Display Next Path
              - Display Previous Path
              - Analyze Current Path
              - Delete Current Path
              - Print Path
              - Return to Menu
```

Figure 3. SCAT Reverse Current Path Display

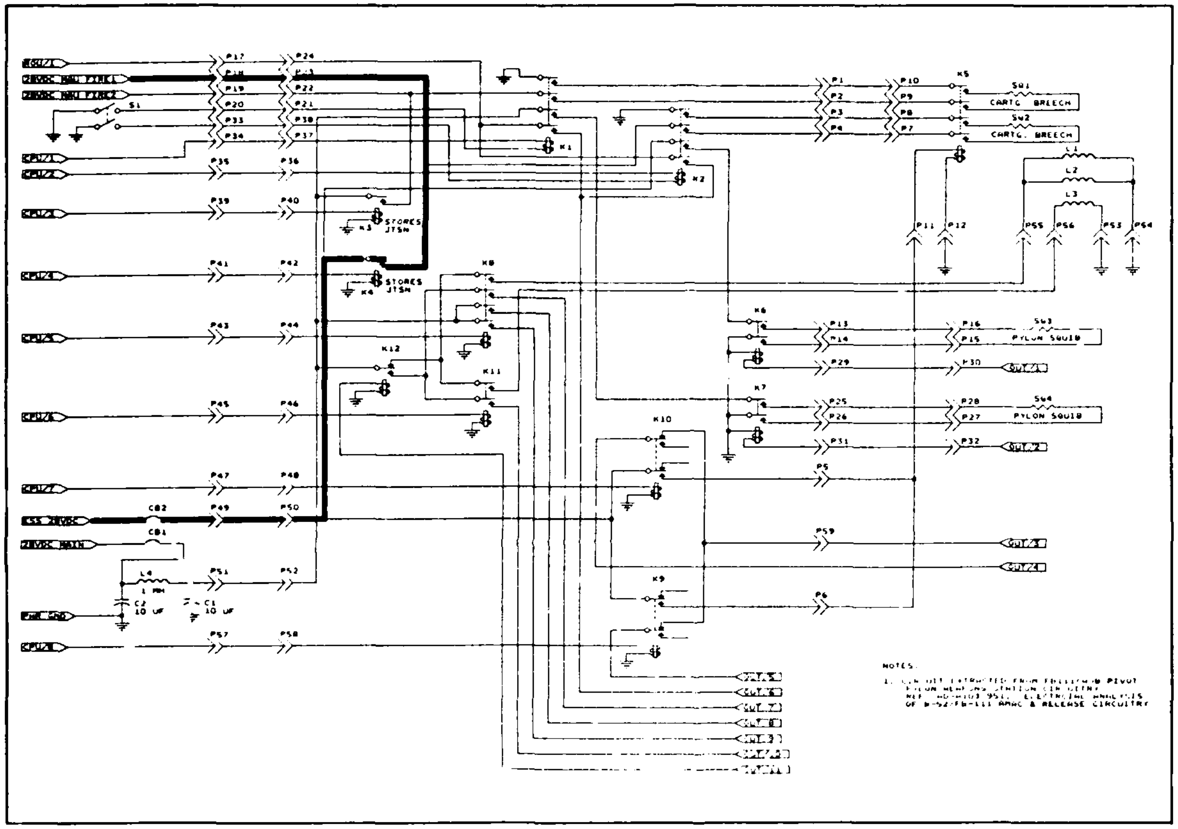
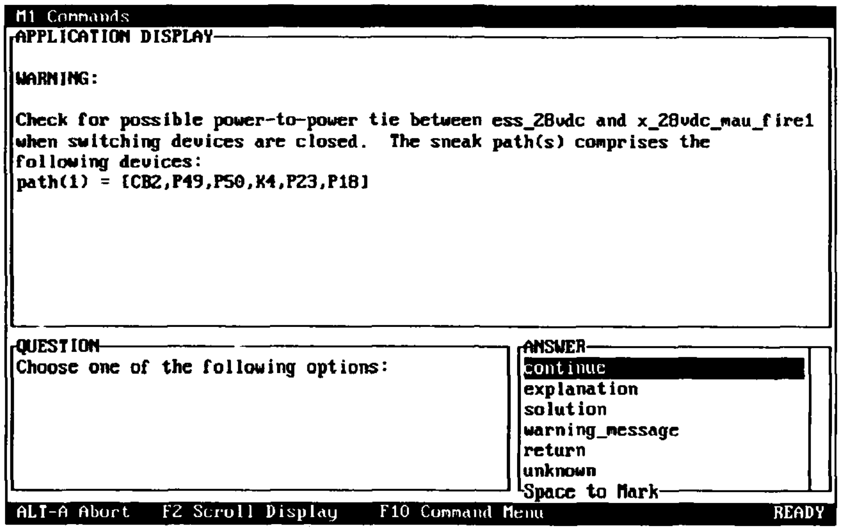


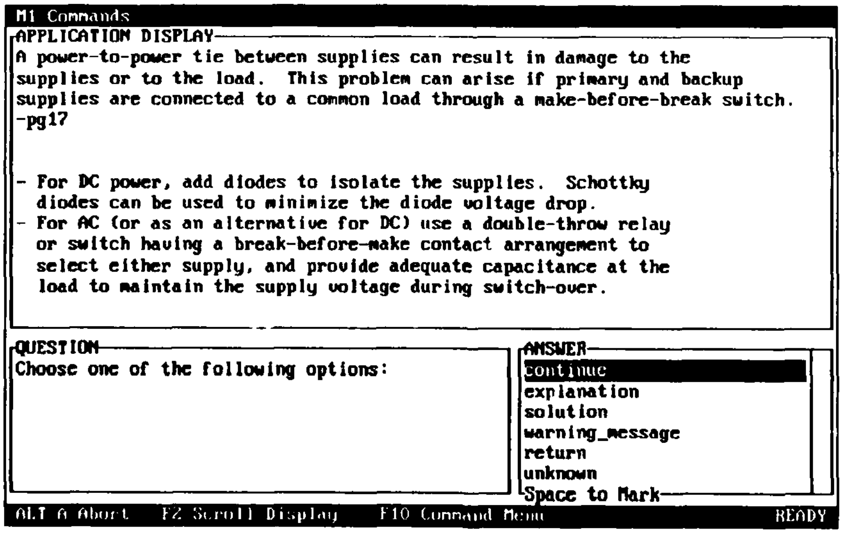
Figure 4. Power Tie in a Weapon Station Circuit



```
WARNING:

Check for possible power-to-power tie between ess_28vdc and x_28vdc_mau_fire1
when switching devices are closed.  The sneak path(s) comprises the
following devices:
path(1) = [CB2,P49,P50,K4,P23,P18]
```

```
Choose one of the following options:
```

Figure 5. SCAT Power Tie Display



```
A power-to-power tie between supplies can result in damage to the
supplies or to the load.  This problem can arise if primary and backup
supplies are connected to a common load through a make-before-break switch.
-pg17


- For DC power, add diodes to isolate the supplies.  Schottky
  diodes can be used to minimize the diode voltage drop.
- For AC (or as an alternative for DC) use a double-throw relay
  or switch having a break-before-make contact arrangement to
  select either supply, and provide adequate capacitance at the
  load to maintain the supply voltage during switch-over.
```

```
Choose one of the following options:
```

Figure 6. Explanation and Solution Messages

APPENDIX A

SCAT USER'S MANUAL

A.1 INTRODUCTION

Sneak Circuit Analysis (SCA) is an established procedure for identifying sneak related problems (sneak paths, sneak timing, sneak labels/indications, design concerns, drawing errors) in electrical circuits. The procedure is specified as Task 205 in MIL-STD-785B where sneak circuits are defined as unintended paths that can cause an undesired function to occur or a desired function not to occur, assuming no component failures[1]. A non-topological version of SCA is specified in MIL-STD-1543B where functional paths and design concerns are addressed.

Standard SCA procedures are highly labor intensive and process input data available only during the latter portion of the development cycle. Systems have been developed for automating the data-formatting portions of the procedure, but these require expensive computer resources (typically large, batch processing systems) and experienced analysts. The Sneak Circuit Analysis Technique (SCAT) overcomes these deficiencies by providing a personal computer based system for real time identification of sneak paths and design concerns early in the development cycle with no prior SCA experience required of the analyst. These features are attained in part by targeting the analysis to identify sneak paths in switching circuits and commonly encountered design concerns related to sneak paths in analog or digital circuits, and in part by focusing the analysis at the assembly or subsystem level rather than the entire system. In this manner, most sneak problems can be identified and corrected by the responsible design engineer in a timely manner.

The automated analysis is based in part on the fact that sneak paths involve circuit branches that conduct current in either direction depending upon the switching state of the circuit. Thus, SCAT searches for these bidirectional branches rather than perform the more complex task of searching for specific topological circuit patterns as done by conventional automated SCA approaches. The analyst's task is also reduced to evaluating the significance of specific sneak paths rather than applying "clue lists" to circuit patterns for identifying the sneak paths.

[1] Sneak software analysis is not addressed here.

Since only a portion of a system is being analyzed at any given time, a feature has been provided to identify sneak paths associated with assembly or subsystem interfaces. SCAT requests the user to identify each interface port of a switching circuit in terms of it being a power input, ground return, or signal I/O. Interfaces to power and ground are labeled as such regardless of whether they respectively go to power and ground directly or through switched or unswitched loads, and they are included within the sneak path search. It is important to note that SCAT is intended to identify sneak conditions early in the design when interfaces may not yet be completely defined. In this manner, problems can be corrected early at minimal cost so that at a later development phase a more conventional SCA can be performed on the entire system and uncover the few, if any, remaining problems.

A.2 THE AUTOMATED PROCEDURE

The automated procedure for performing SCA consists of the following steps:

1. Target critical areas of a system for analysis.

The requirement for performing a SCA should apply to portions of a system considered critical. These subsystems can be identified from the results of other analyses such as FMEA or Fault Tree. Bear in mind that sneak path analysis addresses combinatorial power switching and distribution circuits and that design concern analysis, although applicable to all analog and digital circuitry, is not intended for identifying sneak paths. To insure thoroughness, all external interfaces of the targeted subsystems must be defined in terms of being dedicated power or ground, switched power or ground, or signal lines.

2. Partition the circuitry to be analyzed into manageable segments.

The appropriate size of a segment is a function of the following constraints:

a. The ability of the analyst to understand the detailed operation of the circuit. The analyst must (1) evaluate the operational implications of each reverse current path identified by the sneak path search and (2) respond to SCAT queries concerning circuit timing and the function of circuit components. The size of the circuit must not exceed the analyst's capability to do so.

b. The ability to capture the circuit using OrCAD/SDT. The circuit must be captured using OrCAD/SDT device libraries and editing guidelines specified in section A.5.

c. The ability of SCAT to process the circuit. For typical circuit topologies, sneak path analysis can be performed on circuits containing up to 2,000 components while design concern analysis can be performed on circuits containing up to 300 components.

If circuit partitioning is required, minimize the number of interfaces crossing a partition boundary (see Figure A-1). This can usually be achieved by *functionally partitioning* the circuitry. As before, all interfaces must be defined in terms of being dedicated power or ground, switched power or ground, or signal lines.

3. Generate the EDIF netlist following the procedure described in section A.5.

4. Run SCAT as described in section A.6.

A.3 SYSTEM REQUIREMENTS AND INSTALLATION PROCEDURE

SCAT is a menu driven system consisting of (1) a set of programs written in the C language and (2) knowledge bases developed under the M.1 (Teknowledge, Inc.) expert system shell, all running under MS-DOS and controlled by a DOS batch file program. The user must have a copy of M.1 to run SCAT. Hardware requirements are an 80286 or 386-based personal computer with a minimum of 640K RAM and a 10MB hard disk (an IBM-XT class machine can be used, but performance in terms of circuit size and analysis speed will degrade).

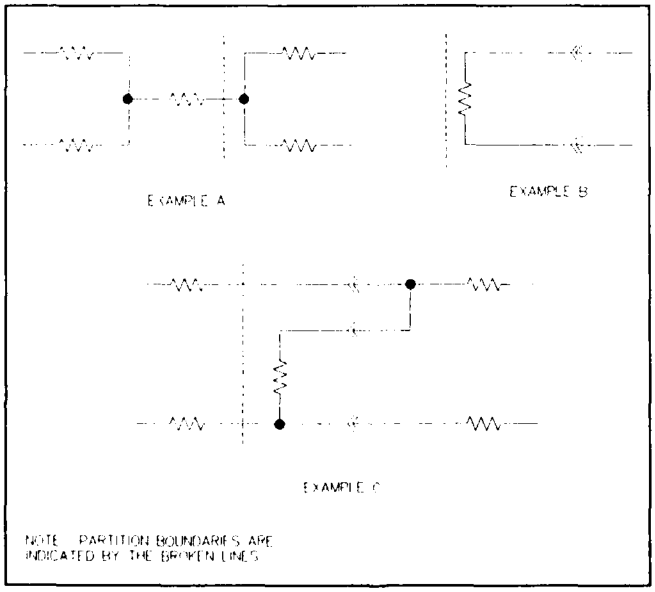

EXAMPLE A
EXAMPLE B
EXAMPLE C
NOTE: PARTITION BOUNDARIES ARE INDICATED BY THE BROKEN LINES

Figure A-1. Proper Circuit Partitioning

Input data for SCAT is a schematic netlist (i.e., a list of device interconnections) formatted in EDIF[2] version 1 1 0 as generated by the OrCAD/SDT III schematic capture tool. OrCAD, a popular, commercially available MS-DOS based CAD package, is not bundled with SCAT.

The SCAT installation procedure is as follows:

1. Insure your computer's system configuration file permits 8 or more files to be open concurrently (i.e., FILES = x where x ≥ 8). For MS-DOS, the default value of the FILES parameter is 8.

2. You must have a copy of M.1 version 2.1 (or a later compatible release) installed on your hard disk in accordance with the vendor's (Teknowledge) installation instructions.

3. Copy all files from the SCAT program floppy disk on to your hard disk in the same directory as M.1.

4. Copy all netlists to be analyzed into the same directory as M.1. Netlists must be generated by OrCAD/SDT III using the "FlatEDIF" data format.

A.4 SCAT OPERATION

This section provides a detailed description of SCAT operation. This material is not required for understanding the operating procedure that appears in section A.6.

A complete list of names of SCAT programs and knowledge bases appears in Table A-1. Documented listings of the source code and knowledge bases appears in volume II of the final report. A list of temporary data files generated by the programs appears in Table A-2. SCAT programs controlled by the DOS batch file SCAT.BAT are shown by the diagram appearing in Figure A-2. The program names appearing in the figure are referred to in the following discussion.

Upon invoking SCAT, the netlist entry screen generated by the knowledge base SCAFILE enables the user to specify a netlist file to be analyzed. At this point, control returns to SCAT.BAT which invokes the C program EDIF2M1. This program reads the specified netlist file and outputs a reformatted version in two files (DEVS.SCA and JOINS.SCA) for use by the M.1 knowledge bases. If the netlist file is not found, SCAT.BAT calls the M.1 file SCAFILEB which displays an appropriate error message and re-requests a netlist file name. Otherwise, SCAT.BAT calls M.1 file SCAMENU which generates the "main menu" for SCAT.

[2] Electronic Data Interchange Format as specified by EIA Interim Standard No. 44. Version 2 0 0 was not available at the time this effort was undertaken.

Table A-1. SCAT Programs, Knowledge Bases and Databases

File Name	File Type/Size	Description
SCAT.BAT	DOS batch file/1K	Controls overall program flow. Invoked by user.
DESIGN	KBase/12.3K	Identifies power/ground paths. Calls PATHS. Called by SCAMENU.
DEVICE	KBase/24.4K	Identifies device-related design concerns from the netlist. Called by DSGNMENU.
DSGNMENU	KBase/5.9K	Function/device design concerns selection. Called by SCAT. Calls FUNCTION or DEVICE.
EDIF2M1.EXE	C Program/19.5 K	Generates DEV.SCA and JOINS.SCA from the netlist. Called by SCAT.
FUNCTION	KBase/46.1K	Identifies function-related design concerns from the netlist. Called by DSGNMENU.
GOODPART.PTH	ASCII/.4K	Data base used by PATHS.
M1.EXE	M1 shell/231.6K	Expert system inference engine.
MODELS.SCA	ASCII/3K	Data base used by SCA and PATHS. Contains models for circuit devices.
PATHS.EXE	C Program/33.9 K	Identifies power-power sneak paths. Called by SCAT.
SCA.CFG	M1 shell/2.4K	Display configuration data.
SCA.EXE	C Program/36K	Identifies sneak paths from the netlist. Deleted parts and source/sink nodes are read from DATA.SCA. Called by SCAT.
SCAFILE	KBase/3.2K	Netlist file selection. Called by SCAT.
SCAFILEB	KBase/3.6K	Reports netlist file selection errors. Called by SCAT.
SCAIN	KBase/14.6K	Generates DATA.SCA. Reports presence of ICs in the netlist. Called by SCAMENU.
SCAMENU	KBase/11.4K	Sneak paths/design concerns selection. Called by SCAT. Calls DESIGN or SCAIN.
SCAOUT	KBase/13.2K	Displays sneak paths. Assists user evaluation. Calls PATHS. Called by SCAT.

Table A-2. SCAT Data Files

	File	
File Name	Type/Size	Description
BIS.SCA	ASCII/**	Sneak paths identified by SCA for SCAOUT.
DATA.MOD	ASCII/*	Generated by SCAMENU. Contains switch, relay, and capacitor models.
DATA.PTH	ASCII/*	Power sources identified by DESIGN for use by PATHS. Also used as a flag file for SCAT to execute PATHS.
DATA.SCA	ASCII/*	Generated by SCAIN. Lists deleted netlist parts and source/sink nodes.
DATABAK.MOD	ASCII/.03K	Initial system data for switch, relay, and capacitor models.
DATABAK.SCA	ASCII/.09K	Initial system data for source/sink nodes.
DELS.SCA	ASCII/*	Generated by SCAOUT and used by SCA. Contains deleted paths.
DEVS.SCA	ASCII/*	Generated by EDIF2M1. Identifies all netlist parts and ports. The file is used by various KBases.
DONE.DSN	ASCII/**	Flag generated by DSGNMENU for SCAT to execute SCAMENU.
DONE.SCA	ASCII/**	Flag generated by SCAMENU for SCAT to terminate and exit to DOS.
FILE.SCA	ASCII/*	Contains netlist file name generated by SCAFILE or SCAFILEB for EDIF2M1 and SCA.
JOINS.SCA	ASCII/*	Generated by EDIF2M1. Identifies circuit nodes in the netlist. The file is used by various KBases.
PATHS.PTH	ASCII/*	Power-power paths identified by PATHS for FUNCTION.
SNEAK.SCA	ASCII/**	Flag generated by SCAIN for SCAT to execute SCA.

* Varies with circuit size.
** Program flag. Memory requirement is negligible.

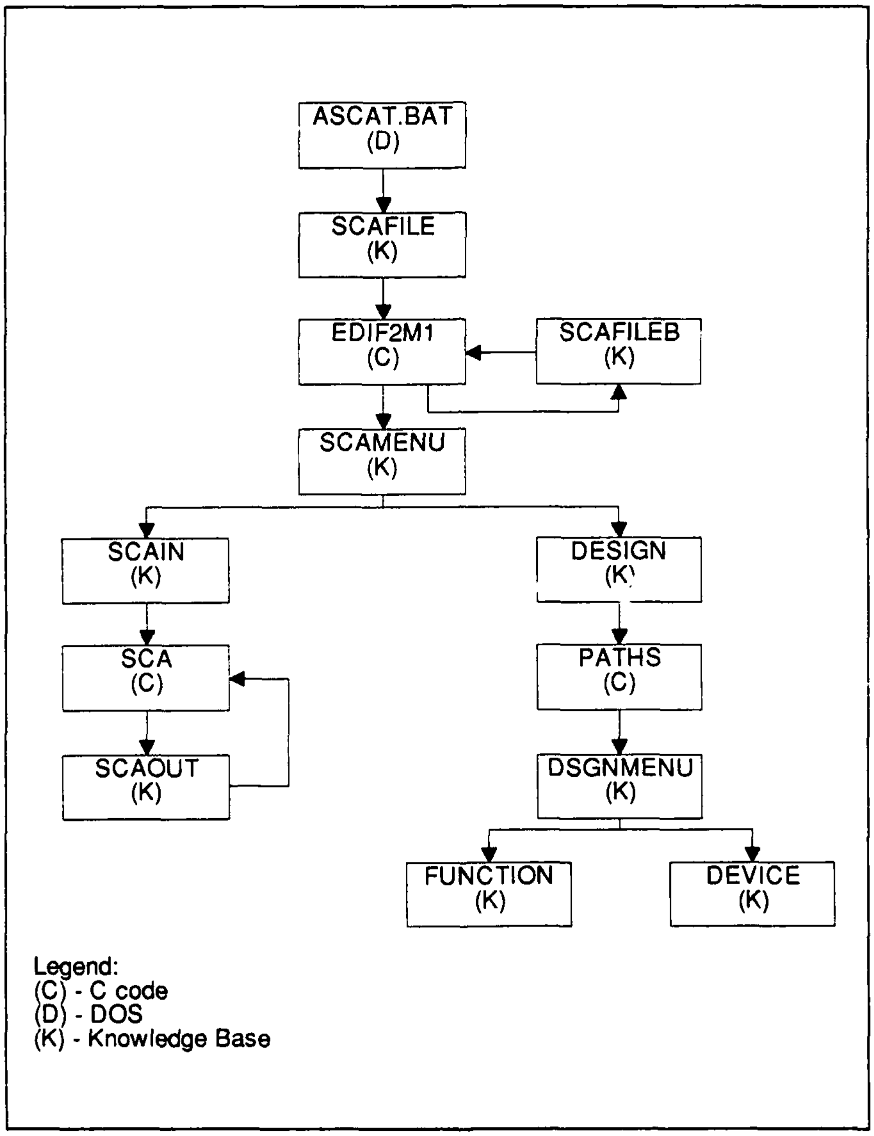


Figure A-2. SCAT Program

The main menu allows the user to modify switch, relay and capacitor models before choosing one of the following types of analyses:

(1) Sneak Path Search -- performed by the C program SCA and the knowledge bases SCAIN and SCAOUT.

(2) Design Concern Analysis -- performed by the C program PATHS and the knowledge bases DESIGN, DSGNMENU, FUNCTION and DEVICE.

Sneak path search applies only to switching circuitry, i.e., circuitry consisting of current interrupting devices (switches, relays, connectors, circuit breakers, fuses). Sneak paths are not traced through integrated circuits; ICs are instead automatically modeled by SCAT as open terminations (i.e., IC leads are treated as open circuits). However, by using OrCAD to edit the schematic, the user can substitute equivalent circuits if he is aware of them.

SCAT searches for potential sneak paths by first identifying all directed, non-circular, topological paths between two user specified nodes; the "source" node (starting point) and the "sink" node (ending point). These paths are then analyzed to identify those that are bidirectional (i.e., capable of conducting current in either direction depending upon the circuit's switching state. Each reverse current path identified is displayed to the user as an ordered list of device reference designators keyed to the circuit schematic. The user traces each reverse current path on the schematic to determine the path's validity (i.e., whether system operation precludes the assumed switching state required by the path) and its significance (i.e., its effect on mission success, personnel safety, equipment damage, etc.).

Reverse current paths are identified from the netlist by the C program SCA. The knowledge base SCAIN enables the user to specify source and sink nodes and to set the switching state of any device. The knowledge base SCAOUT displays the reverse current paths and provides assistance in evaluating their validity. SCAOUT also allows the user to mark any reverse current path as invalid and regenerates the paths to eliminate the marked one and all others solely dependent upon it. A complete analysis requires that the sneak path search be re-run for all combinations of source and sink nodes involving power supplies (sources) and grounds (sinks).

Design Concern Analysis differs from Sneak Path Search in that the former is a highly interactive consultation and can be applied to any analog, digital or combined analog/digital circuitry. The analysis identifies problems associated with (1) circuit configurations involving specific devices (DEVICE knowledge base) and (2) circuit configurations involving circuit functions such as power distribution (FUNCTION knowledge base). Design concerns are implemented as knowledge base rules and comprise the functional guidelines and device guidelines appearing in the guidebook Sneak

Circuit Analysis for the Common Man[3]. When a design concern is encountered, an appropriate message is displayed and the user is given the options of requesting an explanation of the problem and a possible solution. The analysis concludes by returning to the main menu.

A.5 SCHEMATIC AND NETLIST GENERATION

Schematics must be drawn using the graphics and text editing tools and parts symbol libraries supplied by OrCAD/SDT III. The product includes print and plot utilities for generating hardcopies as well as a utility for generating netlists. Completed schematics are saved on disk and may be retrieved for additional editing. A complete description of the product can be found in the OrCAD user's manual and is not presented here.

Before generating a netlist for input to SCAT, the schematic must be checked for the following:

1. All interfaces to external power sources and ground nodes must be labeled using the OrCAD/SDT module port facility.

2. The terminals of all in-circuit voltage sources (e.g., batteries) must be labeled using the OrCAD/SDT module port facility.

3. Any device appearing in the OrCAD/SDT DEVICE library may be used. The devices are shown in Figures A-3 and A-4. In addition, any IC may be used as long as it is referenced by the prefix "U" (see item 4).

4. Schematic reference designations for circuit components must use the default label prefix provided by OrCAD/SDT (e.g., "R" for resistor one, "K" for relay, "U" for IC, etc.). Refer to Figures A-3 and A-4 for the default reference designator of each device.

5. Any labeled power, ground or signal path may be specified as a source or sink for Sneak Path Analysis. This may be done (1) while running SCAT by specifying the path name, or (2) while editing the schematic by labeling the desired module port as "SRC" or "SNK".

The OrCAD/SDT schematic error checking utility, ERC, can be used to check circuit connectivity for shorts between outputs, inputs with no driving source, unconnected pins and other common wiring errors.

The netlist is generated using the OrCAD/SDT NETLIST utility. The special format "FlatEDIF" must be specified when invoking the utility.

[3] The guidebook's *design rules* for avoiding sneaks during design were not implemented during this effort because they apply functionally to the overall circuitry and are therefore much more difficult to automate.

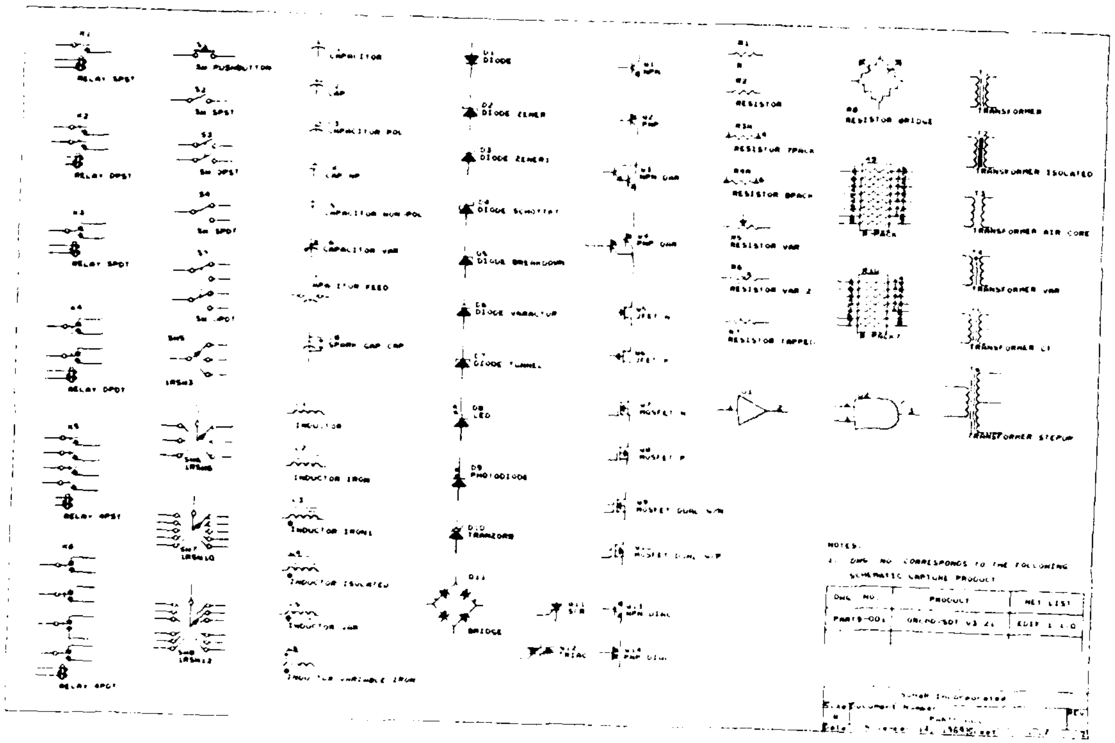


Figure A-3. Schematic Symbols for SCAT Net Lists

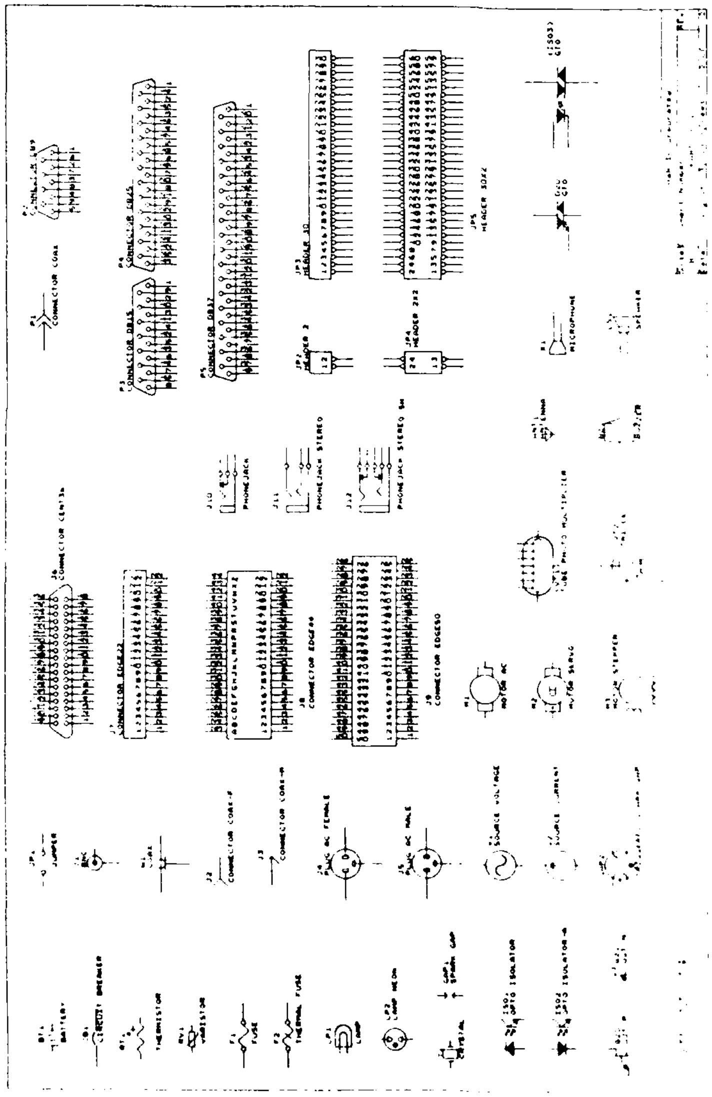


Figure A-4. Additional Schematic Symbols

A.6 SCAT OPERATING PROCEDURE

Note: In the following procedure, data to be entered (i.e., entry followed by carriage return) appear enclosed in chevrons (i.e., <data>). Entries must be typed lower case. Do not type the chevrons. Entries to be selected are enclosed in quotes (i.e., "selection"). Entries are selected by using arrow keys to highlight the selection and pressing ENTER. The selection "unknown" is not operative.

1. Enter <scat>.

   A netlist file name will be requested.

2. Enter the name of the netlist file for the circuit to be analyzed. Use lower case text.

   The file will be retrieved and pre-processed. This will typically take 5-10 seconds. If the file is not found, a bad-file message will appear and a netlist file name re-requested. Otherwise, the main menu will appear and the user is requested to select the type of analysis to be performed.

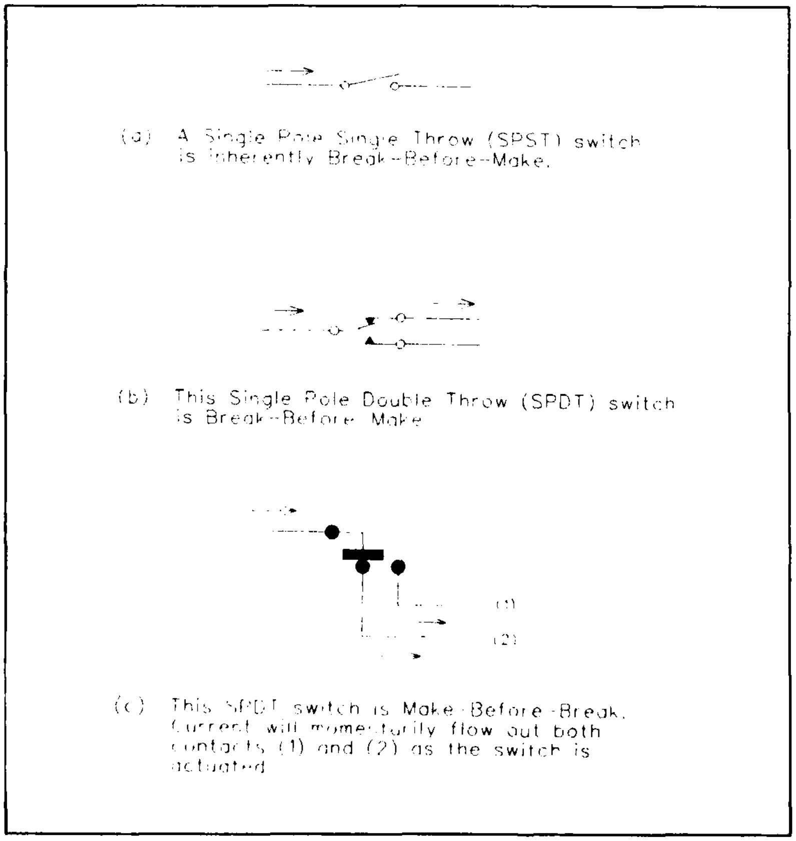
3. Before proceeding to select Sneak Path Analysis ("sneak") or Design Concerns Analysis ("design"), the user has the option of modifying switch/relay and capacitor models to agree with their engineering specifications or usage. The contact arrangement for multiple-throw switches and relays (see Figure A-5) can be specified as Break-before-Make or Make-before-Break. In the former (the default case), when the switch or relay is toggled, the newly selected path is established after the old path has been opened. In the latter, there is some overlap for a short period (typically a few milliseconds) during which time the newly selected path and the old path exist concurrently. Models corresponding to these two configurations can be specified for switches and relays either individually or globally.

   Capacitor terminals can be modeled as being unconnected to each other (i.e., OPEN) or connected together (i.e., SHORTed). The former (the default case) applies to paths involving DC currents while the latter applies to AC or transient current paths. As before, models corresponding to these two configurations can be specified for capacitors either individually or globally.

4. Option 1: Select "sneak" paths.

   If the netlist includes ICs, the user is informed that sneak path processing will treat paths to these devices as open circuits. The user has the option of continuing with the analysis or returning to the main menu.

   If both a source node and sink node have not been specified in the schematic, the user is requested for these data.



(a) A Single Pole Single Throw (SPST) switch is inherently Break-Before-Make.

(b) This Single Pole Double Throw (SPDT) switch is Break-Before Make

(c) This SPDT switch is Make-Before-Break. Current will momentarily flow out both contacts (1) and (2) as the switch is actuated

Figure A-5. Types of Switching Devices

When source and sink nodes have been specified and if there are no ICs, or if there are and the user chooses to continue, the sneak path input data menu appears. The user may choose to:

4.1 Delete parts to simulate OPENed switches, remove redundant paths (thereby reducing processing time), delete paths for some other arbitrary reason. A list of the deleted parts and the reason for their deletion is maintained by the program.

4.2 Undelete previously deleted parts.

4.3 Change the source node for the analysis. If a node is labeled SRC on the schematic, it is the default source.

4.4 Change the sink node for the analysis. If a node is labeled SNK on the schematic, it is the default sink.

4.5 Execute the sneak path search.

Sneak path processing will commence. Processing time for small circuits (<u>i.e.</u>, short netlists) is a few seconds; larger circuits require more time. The reverse current path menu will appear when processing is concluded. Each path is identified by a list of device names corresponding to those in the circuit schematic that lie on the sneak path between the source and sink nodes. Devices that lie on the bidirectional portion of the path are prefixed by an "*". As an aid for cross-referencing the path list to the schematic, relays are listed with either a "-S" suffix to indicate a switching contact or a "-C" to indicate a coil. The analyst can trace the path on a copy of the schematic to facilitate its evaluation. The following options are available from the reverse current path menu:

4.5.1 Display the next path. Paths are consecutively numbered for reference.

4.5.2 Re-display the previous path. The path queue is circular.

4.5.3 Mark a reverse current path for deletion. This is required if user determines the path to be operationally impossible (<u>e.g.</u>, due to forbidden switching states).

4.5.4 Unmark a marked path. This option is available only when a path has been marked.

4.5.5 Display deleted paths. This option is available only when one or more paths have been deleted. The deleted path menu is similar to the reverse current path menu.

4.5.6 Regenerate paths. This option is available only when a path is marked for deletion or a deleted path is marked to be undeleted.

4.5.7 Print the path currently displayed (for hardcopy reference).

4.5.8 Request computer-aided analysis of the sneak path. The system will prompt the user for basic, circuit-related information necessary to evaluate the significance of the sneak path in terms of inhibiting desired functions or causing undesired functions. If as a result of the analysis the reverse current path is declared to not be a sneak path, the path will be automatically marked for deletion.

4.5.9 Return to the main menu.

The sneak path search should be repeated for all applicable source/sink pairs. These include each instance of the following combinations:

SOURCE	SINK
+ or - DC power input	corresponding DC return
+ DC power input	- DC power input
AC power input	corresponding AC return

5. Option 2: Select "design" concerns analysis.

The following messages will appear.

5.1 Power/ground message. The analysis requires that power and ground paths be unambiguously identified. Automatic identification of the paths is initially attempted. The user is then asked to validate the power and ground listings and correct them if necessary. Additions or deletions are made by entering power (or ground) names one at a time. As described in the message, the names must appear as labels on the schematic and must be entered in lower case text. Spaces within the name must be replaced by underscores. Names prefaced by a number must be prefixed by "x_".

5.2 Circuit type message. The user must designate the circuitry being analyzed as either analog, digital, or both.

Prior to continuing, the currently identified power and ground nodes and the circuit type are displayed along with an option to modify them. If the design concerns analysis is repeated, this summary screen is displayed in place of the power/ground and circuit type screens.

5.3 Design concerns type message. The analysis is divided into two parts, functional guidelines and device guidelines, to facilitate focusing on specific attributes of the circuit for analysis (or re-analysis). The user is requested to specify the analysis type.

From this point on, queries will arise as the system attempts to identify design concerns. The user must respond to each query for the analysis to continue. Queries addressing specific devices reference those devices by their schematic labels. Where switch or relay contacts are referenced, the specific terminals being ad.ı.essed are indicated along with the device label (e.g., K1-common1). The terminal identifiers are the same as those appearing in the OrCAD device library.

As each design concern is identified, an appropriate message is displayed. The user may request an explanation of the design concern, a possible solution, or re-display of the original message. The user is given an option for printing out identified design concerns at the conclusion of each guideline session.

A.7 SCAT APPLICATION EXAMPLE

Note: In the following procedure, data to be entered appear enclosed in chevrons (i.e., <data>). Entries must be typed lower case, followed by a carriage return. Do not type the chevrons. Entries to be selected are enclosed in quotes (i.e., "selection"). Entries are selected by using arrow keys to highlight the selection and pressing ENTER. The selection "unknown" is not operative. Figures for this section appear at the end of this section.

The example references the schematic DEMO (Figure A-6). SCAT will be used to identify a sneak path (shown highlighted in Figures A-12 and A-14) and a power-to-power tie (shown highlighted in Figure A-28). Entry of specific switch contact timing configurations (i.e., Make-Before-Break or Break-Before-Make) will also be demonstrated. Figure A-29 depicting SCAT program flow, references the screens described in the following example.

1. Install SCAT onto hard disk as described in the SCAT Installation Procedure.

2. At the DOS prompt, enter <**SCAT**>. The netlist entry screen (Figure A-7) will appear.

3. At the netlist entry screen, enter <**demo.net**>. The main menu (Figure A-8) will appear.

4. At the main menu, select "**sneak**". The IC message (Figure A-9) will appear.

5. At the IC message, select "**continue**". The sneak input data menu (Figure A-10) will appear.

6. At the sneak input data menu, select "**execute**". The first reverse current path will appear.

7. At the reverse current paths menu (Figure A-11), observe Path 1 data. The path number appears at the top line of the APPLICATION DISPLAY window. Observe that a total of two reverse current paths were identified. A system reference number for the path also appears. Since no parts were deleted, the DELETED PARTS data, appearing below the path listing, are all empty (i.e., [ ]).

   Path 1 is [K1-S,*S2,K1-C]. The path, shown highlighted in Figure A-12, comprises the source (labeled SRC on the schematic), a switch contact on relay K1, the switch S2, the coil of relay K1, and ground sink (labeled SNK on the schematic). The source and sink nodes do not explicitly appear in the path list but are implied.

   Select "**next**" and "**previous**" and observe the path number change. An evaluation of the two reverse current paths identified reveals the critical one to be Path 1. The path permits source current flowing through relay K1 (when de-energized) and through switch S2 (when enabled) to also flow through the coil of K1, thus energizing the relay. This in turn will open the K1 contact, de-energizing the coil and starting an oscillatory sequence of events. Note that current through S2 flows opposite to that implied by Path 2: [Q1,*S2,LP2] (see Figures A-13 and A-14). Hence, S2 is bi-directional and is prefaced by an asterisk.

   Select "**return**". The main menu will appear.

8. At the main menu, select "**design**". The power source menu (Figure A-15) will appear.

9. At the power source menu, select "**continue**". The ground list menu (Figure A-16) will appear.

10. At the ground list menu, select "**continue**". The circuit type menu (Figure A-17) will appear.

11. Answer the circuit type query "**both**". The summary of design parameters (Figure A-18) will be displayed along with options to modify any of them.

12. Select "**continue**". The design concern type menu (Figure A-19) will appear.

13. At the design concern type menu, select "**functional_guideln**". The function-oriented design concern analysis will commence.

14. A query will appear regarding power and ground connectors. Select "**no**".

15. A warning regarding a possible shock hazard at the ground pin will appear. Select "**explanation**". An explanation of the concern will appear. The page number at the end of the explanation references the page in the report SCA for the Common Man where additional information regarding this concern can be found.

16. Select "**solution**". A solution for the concern will appear. Note that as the text appears, earlier messages scroll off the APPLICATION DISPLAY window. These earlier messages can be retrieved by pressing function key F2 and using the arrow keys to scroll back through the text. To continue the analysis, F2 scrolling must be disabled by pressing the ESCAPE key.

17. Select "**warning_message**". The original warning message is repeated.

18. Select "**continue**". A warning regarding switching devices in ground paths appears.

19. Select "**return**" (where available as a choice) or press ALT-A ("Alt" key and the letter "A") at any time to interrupt the analysis and return to the design concern type menu. Alternatively, the remaining functional design concerns may be viewed by repeatedly selecting "**continue**" until no further concerns have been identified. At that point, a hardcopy of the identified design concerns may be requested, or "**return**" may be selected to return to the design concern type menu.

20. At the design concern type menu (Figure A-19), select "**device_guideln**". The device-oriented design concern analysis will commence.

21. A query regarding loads of specific trensistors will appear. Select "**yes**".

22. A query regarding interruption of rower at the collector terminal of specific transistors will appear. Select "**q3**" then select "**q2**".

23. A warning regarding a possible sneak path through the transistor will appear. The choices "explanation", "solution", "warning_message", "continue" and "return" may be selected as before.

24. Return to the design concern type menu by either selecting "**return**" (where available as a choice), pressing ALT-A, or repeatedly answering queries and selecting "**continue**" until the analysis concludes and then selecting "**return**".

25. At the design concern type menu, select "**return**". The main menu will appear.

At this point, the basic example concludes. Select "**exit_program**" to exit to DOS. Otherwise, proceed to step 26 to continue this example.

26. At the main menu (Figure A-8), select "**model_switch**". The switch/relay model menu (Figure A-20) will appear. Select "**append_M**". A prompt regarding specifying specific switches as Make-Before-Break will appear.

27. Enter <**s3**> in response to the prompt (Figure A-21). Note the switch entry displayed in the DISPLAY window.

Enter <**k1**>. Enter <**done**> to return to the main menu. Again, note the switch entry appearing within the message in the DISPLAY window (Figure A-22).

28. At the main menu, select "**sneak**". The IC message will appear.

29. At the IC message, select "**continue**". The sneak input data menu will appear.

30. At the sneak input data menu, select "**execute**". The system will search the netlist for reverse current paths. The sneak paths screen will appear.

31. At the sneak paths screen, observe that four reverse current paths have now been identified. The additional two paths are due to relay k1 and switch s3 having been modeled as Make-before-Break.

32. Select "**previous**". Path 4, [Q1,*S2,K1-S,LP1], will appear (Figure A-23).

33. Select "**analyze**". A query regarding permissibility of simultaneous switching will appear. Select "**no**".

34. The system will conclude that reverse current Path 4 does not present a sneak problem and will mark the path for deletion (Figure A-24). This conclusion can be overridden by selecting the "**undelete**" option. Select "**regenerate_paths**".

35. The system will search the netlist for reverse current paths under the constraint that the marked path is not a sneak path. Observe that only two paths were found (Figure A-25). The first of the four previous paths, [K1-S,LP2], is no longer a reverse current path due to the deletion of Path 4. Select "**deleted_paths**".

36. The deleted paths screen will appear (Figure A-26). The options "**next**" and "**previous**" can be used to view deleted paths when more than one exists. The option "**undelete**" can be selected to mark a deleted path as a valid sneak path. Select "**paths**" to return to sneak paths menu.

37. Select "**return**". The main menu will appear.

38. At the main menu, select "**design**". A summary of the currently selected design concern analysis parameters will appear.

39. Select "**continue**". The design concern type menu will appear.

40. At the design concern type menu, select "**functional_guideln**". The function-oriented design concern analysis will commence.

41. The first function-oriented design concern will now be a warning regarding a power-to-power tie (Figure A-27). The corresponding sneak path appears in the message and may be traced on the circuit schematic (see Figure A-28). Note that this warning did not appear earlier when switch S3 and relay K1 were modeled as Break-Before-Make. As before, "**explanation**", "**solution**" or "**warning_message**" may be selected.

42. Return to the design concern type menu by either selecting "**return**" (where available as a choice), pressing ALT-A, or repeatedly answering queries and selecting "**continue**" until the analysis concludes and then selecting "**return**".

43. At the design concern type menu, select "**return**". The main menu will appear.

44. At the main menu, select "**exit_program**". The DOS prompt will appear.

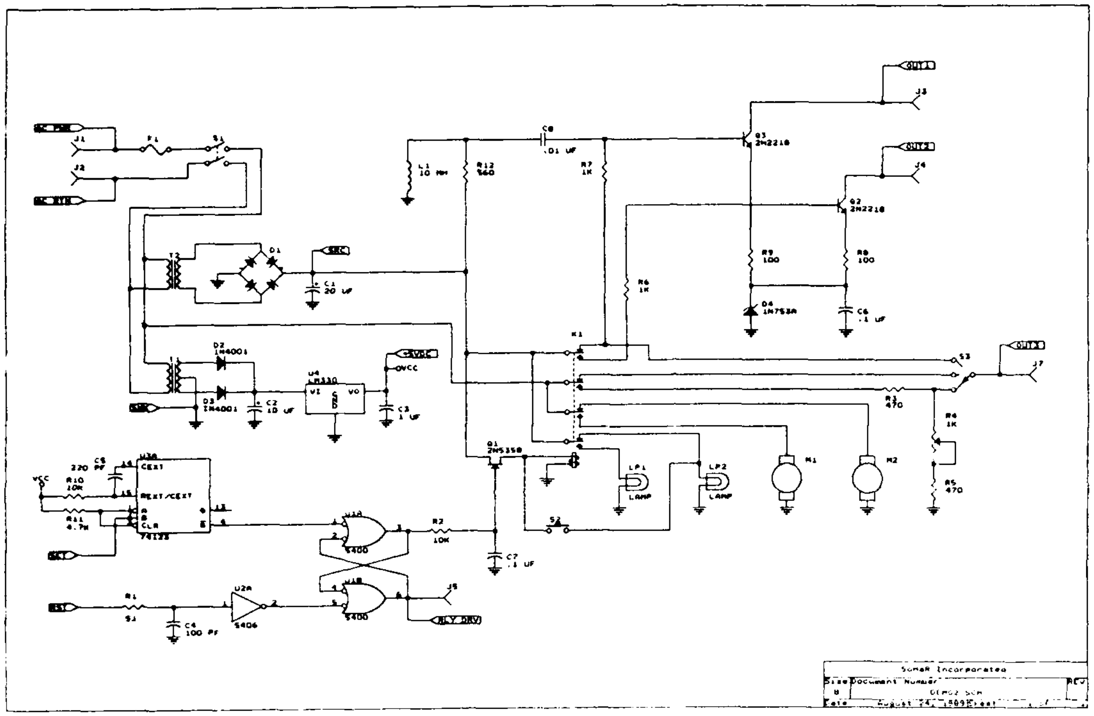
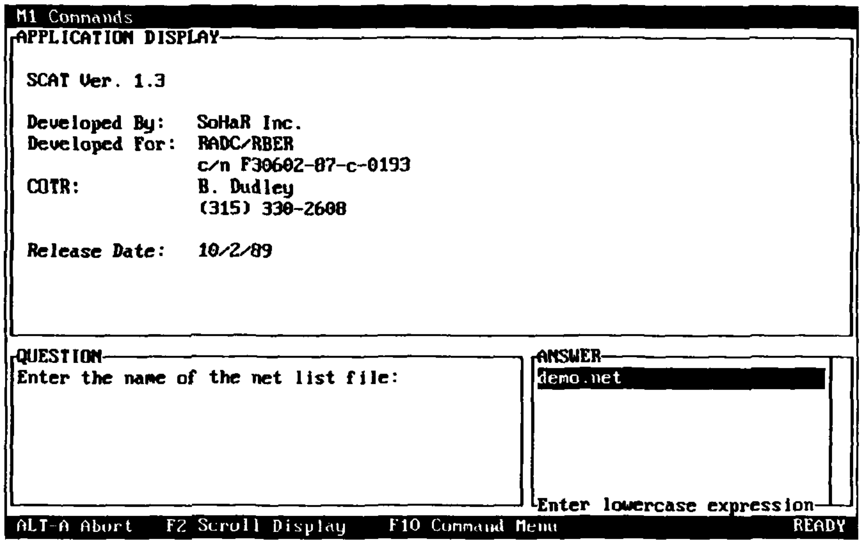
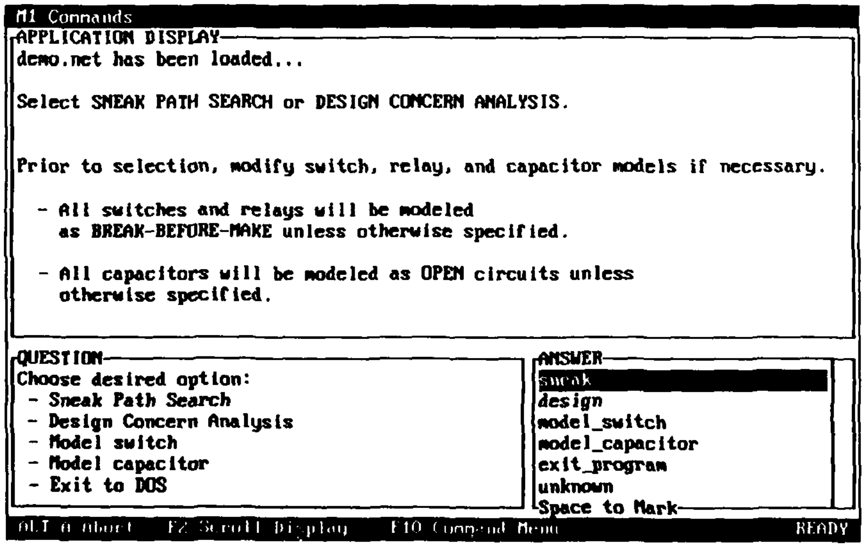


Figure A-6. Schematic of the Example Circuit



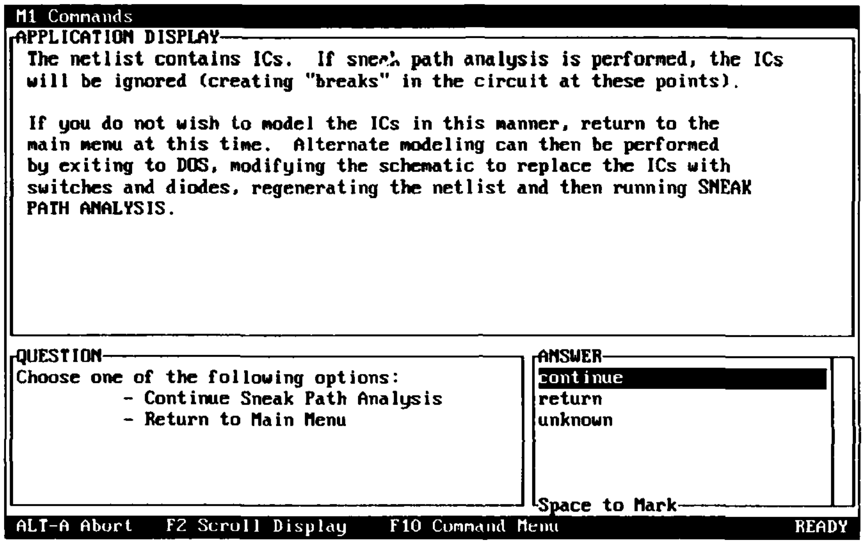
```
M1 Commands
APPLICATION DISPLAY

 SCAT Ver. 1.3

 Developed By:   SoHaR Inc.
 Developed For:  RADC/RBER
                 c/n F30602-87-c-0193
 COTR:           B. Dudley
                 (315) 330-2608

 Release Date:   10/2/89

QUESTION                                           ANSWER
Enter the name of the net list file:               demo.net

                                                   Enter lowercase expression
ALT-A Abort   F2 Scroll Display   F10 Command Menu                     READY
```

Figure A-7. Netlist Entry Screen



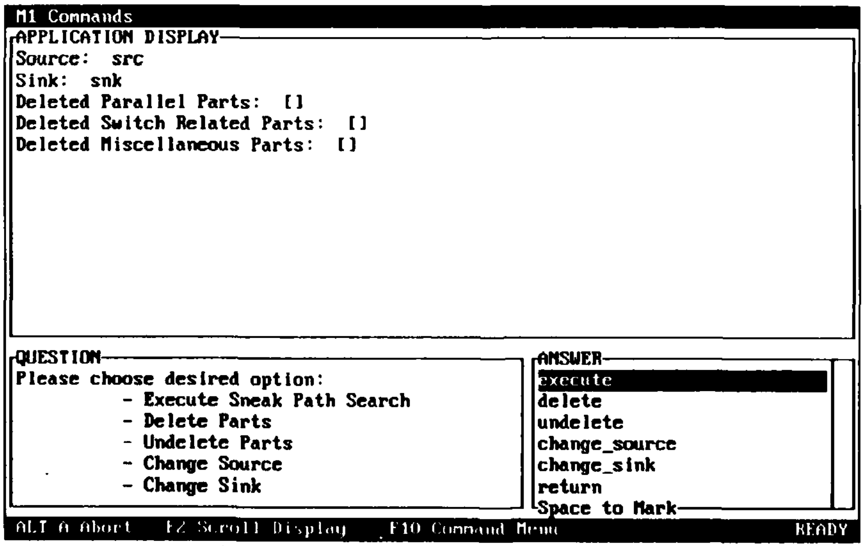
```
M1 Commands
APPLICATION DISPLAY
demo.net has been loaded...

Select SNEAK PATH SEARCH or DESIGN CONCERN ANALYSIS.


Prior to selection, modify switch, relay, and capacitor models if necessary.

  - All switches and relays will be modeled
    as BREAK-BEFORE-MAKE unless otherwise specified.

  - All capacitors will be modeled as OPEN circuits unless
    otherwise specified.

QUESTION                                           ANSWER
Choose desired option:                             sneak
 - Sneak Path Search                               design
 - Design Concern Analysis                         model_switch
 - Model switch                                    model_capacitor
 - Model capacitor                                 exit_program
 - Exit to DOS                                     unknown
                                                   Space to Mark
ALT-A Abort   F2 Scroll Display   F10 Command Menu                     READY
```

Figure A-8. Main Menu



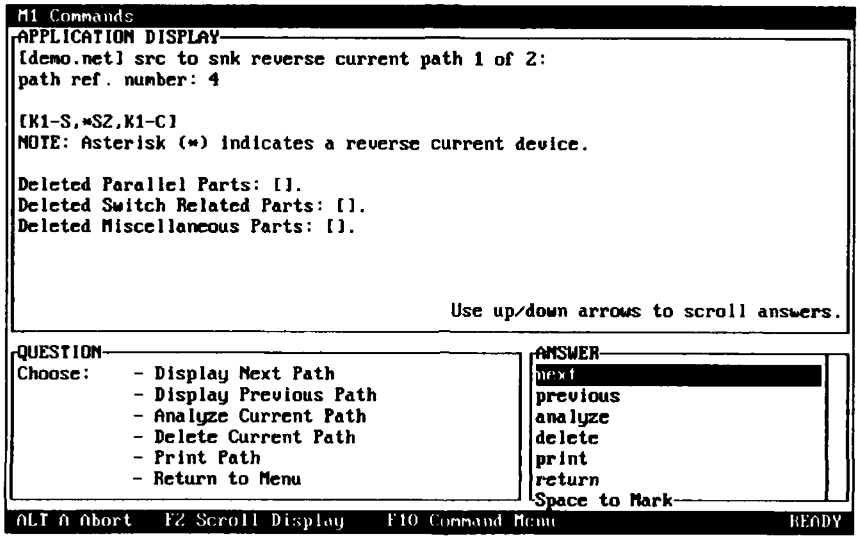
```
M1 Commands
APPLICATION DISPLAY
 The netlist contains ICs.  If sneak path analysis is performed, the ICs
 will be ignored (creating "breaks" in the circuit at these points).

 If you do not wish to model the ICs in this manner, return to the
 main menu at this time.  Alternate modeling can then be performed
 by exiting to DOS, modifying the schematic to replace the ICs with
 switches and diodes, regenerating the netlist and then running SNEAK
 PATH ANALYSIS.

QUESTION                                         ANSWER
Choose one of the following options:             continue
          - Continue Sneak Path Analysis         return
          - Return to Main Menu                  unknown

                                                 Space to Mark
ALT-A Abort   F2 Scroll Display   F10 Command Menu                READY
```

Figure A-9. IC Message



```
M1 Commands
APPLICATION DISPLAY
Source:  src
Sink:  snk
Deleted Parallel Parts:  []
Deleted Switch Related Parts:  []
Deleted Miscellaneous Parts:  []

QUESTION                                         ANSWER
Please choose desired option:                    execute
          - Execute Sneak Path Search            delete
          - Delete Parts                         undelete
          - Undelete Parts                       change_source
          - Change Source                        change_sink
          - Change Sink                          return
                                                 Space to Mark
ALT-A Abort   F2 Scroll Display   F10 Command Menu                READY
```

Figure A-10. Sneak Input Data Menu



```
M1 Commands
APPLICATION DISPLAY
[demo.net] src to snk reverse current path 1 of 2:
path ref. number: 4

[K1-S,*S2,K1-C]
NOTE: Asterisk (*) indicates a reverse current device.

Deleted Parallel Parts: [].
Deleted Switch Related Parts: [].
Deleted Miscellaneous Parts: [].


                                        Use up/down arrows to scroll answers.
QUESTION                                            ANSWER
Choose:     - Display Next Path                     next
            - Display Previous Path                 previous
            - Analyze Current Path                  analyze
            - Delete Current Path                   delete
            - Print Path                            print
            - Return to Menu                        return
                                                    Space to Mark
ALT A Abort   F2 Scroll Display   F10 Command Menu                READY
```

Figure A-11. Reverse Current Path 1

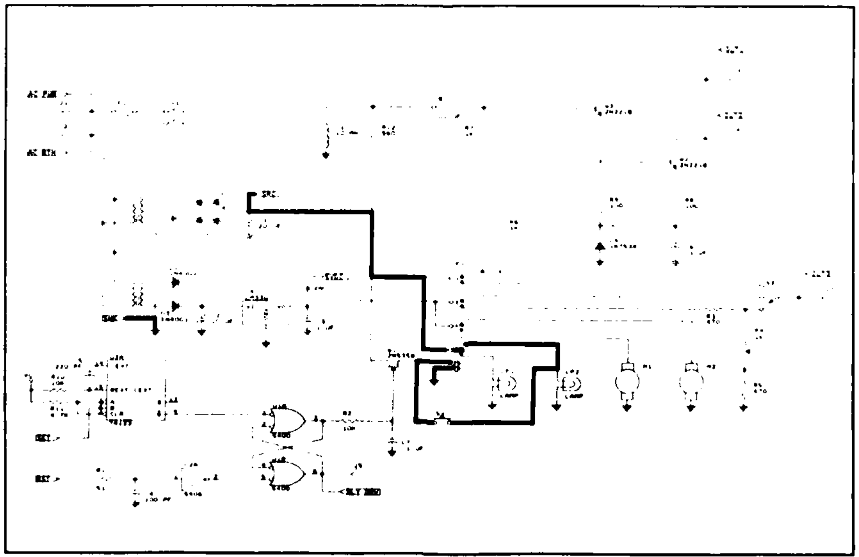
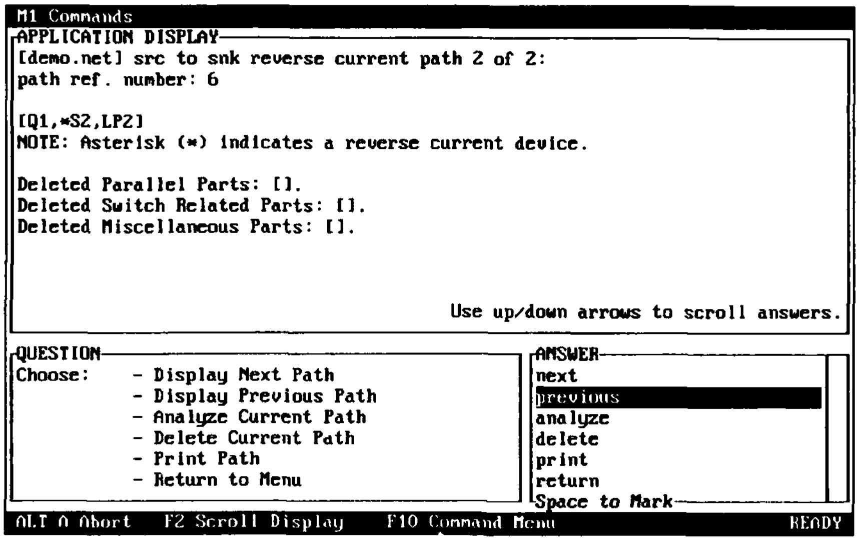


Figure A-12. Path 1 Highlighted on Schematic



```
M1 Commands
APPLICATION DISPLAY
[demo.net] src to snk reverse current path 2 of 2:
path ref. number: 6

[Q1,*S2,LP2]
NOTE: Asterisk (*) indicates a reverse current device.

Deleted Parallel Parts: [].
Deleted Switch Related Parts: [].
Deleted Miscellaneous Parts: [].



                                      Use up/down arrows to scroll answers.

QUESTION                                         ANSWER
Choose:     - Display Next Path                  next
            - Display Previous Path              previous
            - Analyze Current Path               analyze
            - Delete Current Path                delete
            - Print Path                         print
            - Return to Menu                     return
                                                 Space to Mark
ALT A Abort    F2 Scroll Display    F10 Command Menu              READY
```

Figure A-13. Reverse Current Path 2

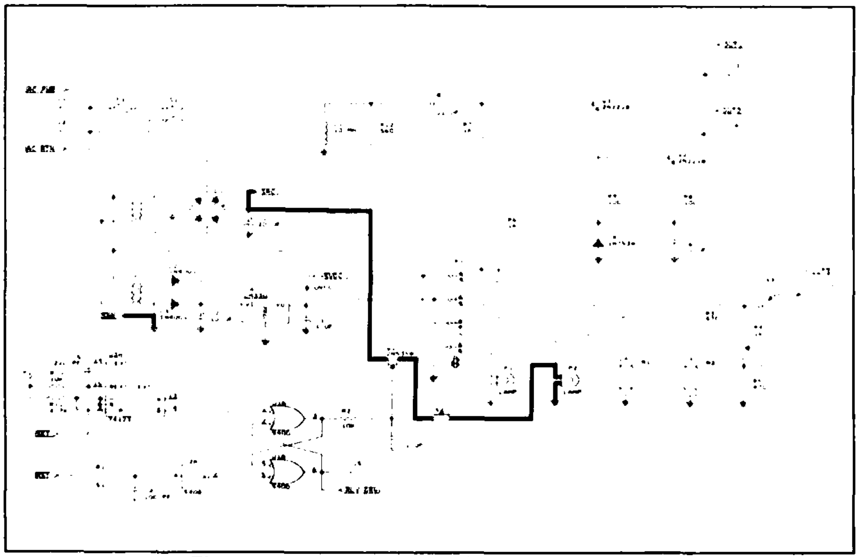
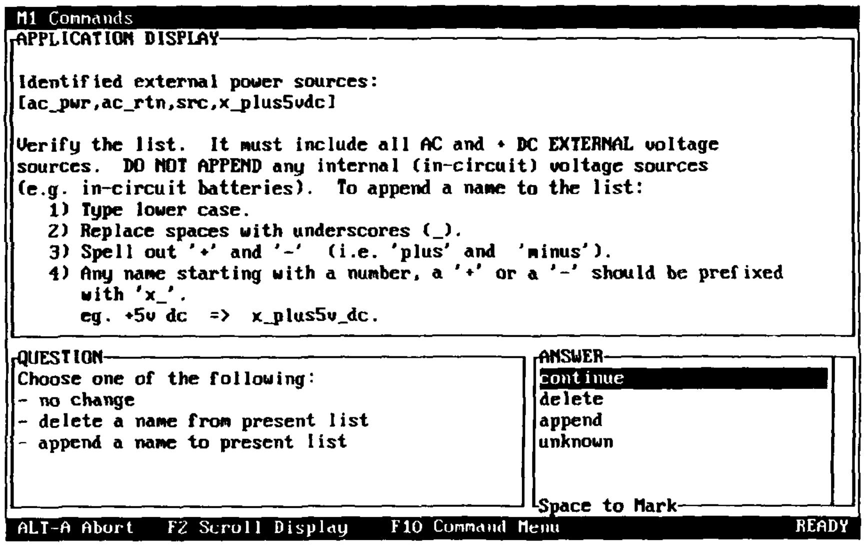
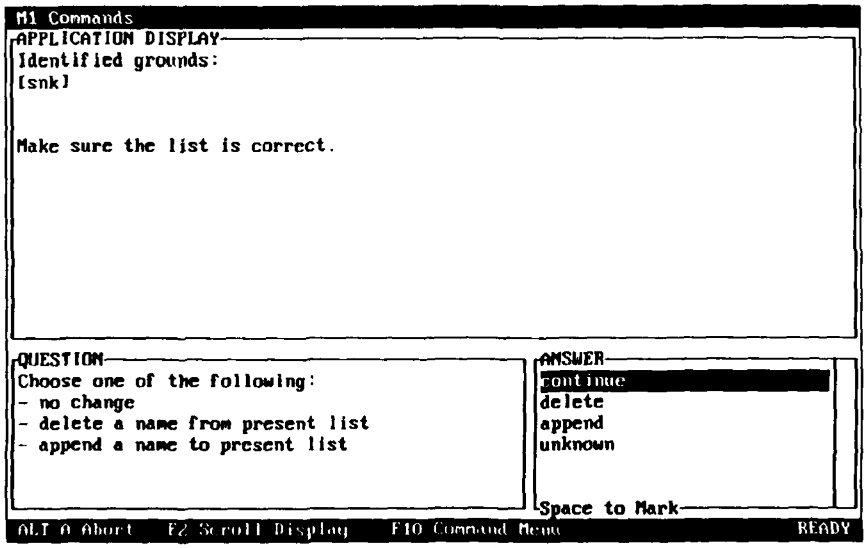


Figure A-14. Path 2 Highlighted on Schematic



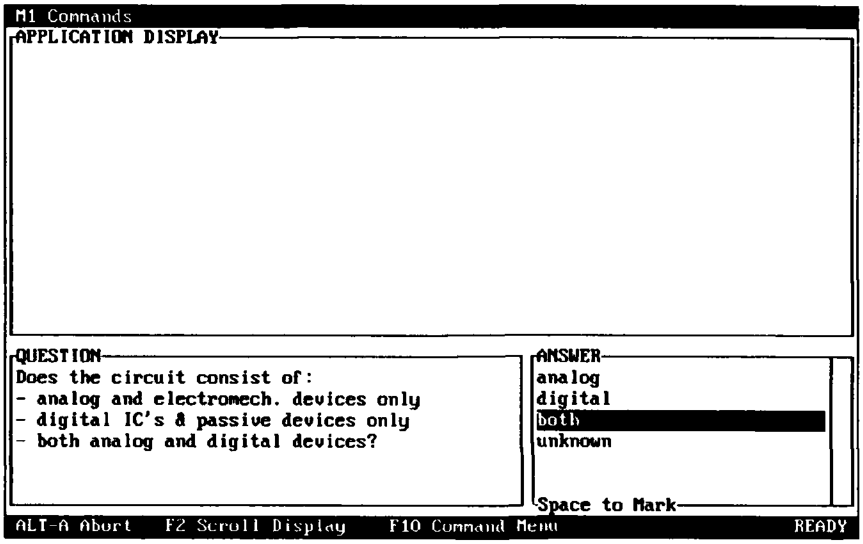
```
M1 Commands
APPLICATION DISPLAY

Identified external power sources:
[ac_pwr,ac_rtn,src,x_plus5vdc]

Verify the list.  It must include all AC and + DC EXTERNAL voltage
sources.  DO NOT APPEND any internal (in-circuit) voltage sources
(e.g. in-circuit batteries).  To append a name to the list:
   1) Type lower case.
   2) Replace spaces with underscores (_).
   3) Spell out '+' and '-'  (i.e. 'plus' and  'minus').
   4) Any name starting with a number, a '+' or a '-' should be prefixed
      with 'x_'.
      eg. +5v dc  =>  x_plus5v_dc.

QUESTION                                         ANSWER
Choose one of the following:                     continue
- no change                                      delete
- delete a name from present list                append
- append a name to present list                  unknown

                                                 Space to Mark
ALT-A Abort   F2 Scroll Display   F10 Command Menu              READY
```

Figure A-15. Power Source Menu



```
M1 Commands
APPLICATION DISPLAY
Identified grounds:
[snk]


Make sure the list is correct.

QUESTION                                         ANSWER
Choose one of the following:                     continue
- no change                                      delete
- delete a name from present list                append
- append a name to present list                  unknown

                                                 Space to Mark
ALT A Abort   F2 Scroll Display   F10 Command Menu              READY
```

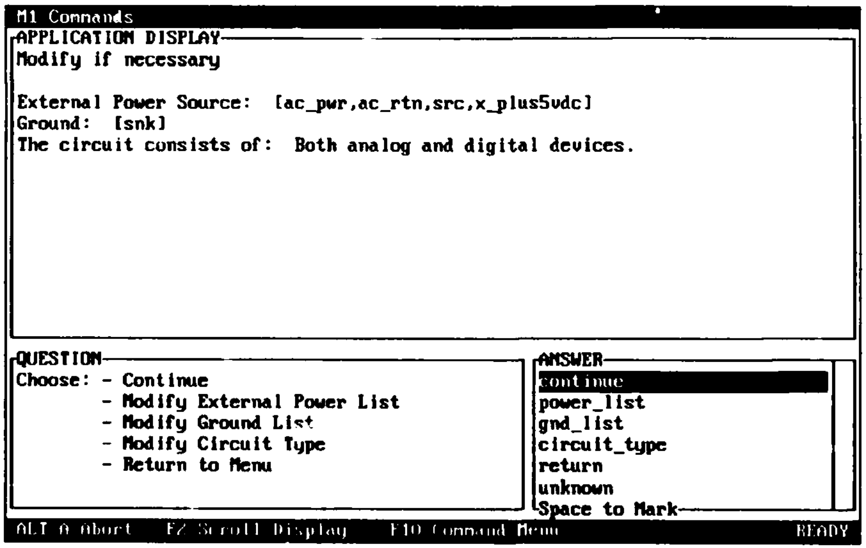
Figure A-16. Ground List Menu



```
Does the circuit consist of:
- analog and electromech. devices only
- digital IC's & passive devices only
- both analog and digital devices?
```

```
analog
digital
both
unknown
```

Figure A-17. Circuit Type Menu

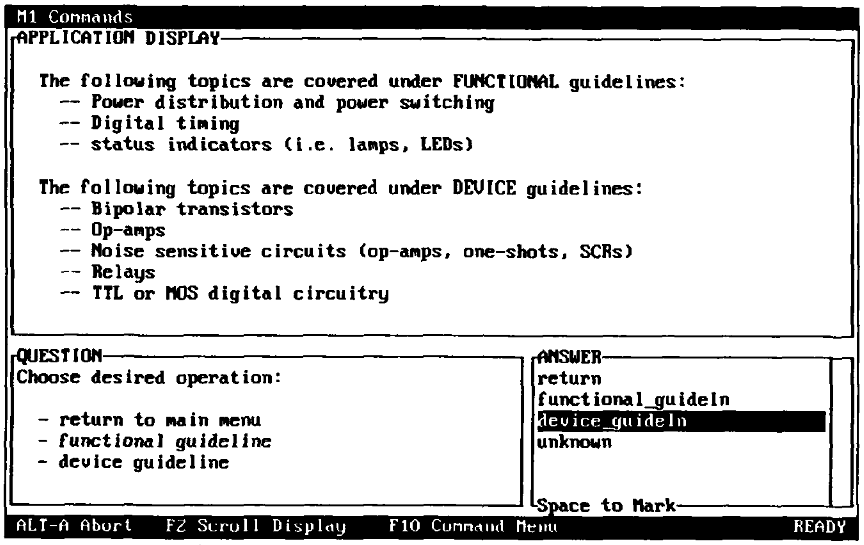


```
Modify if necessary

External Power Source:  [ac_pwr,ac_rtn,src,x_plus5vdc]
Ground:  [snk]
The circuit consists of:  Both analog and digital devices.
```

```
Choose: - Continue
        - Modify External Power List
        - Modify Ground List
        - Modify Circuit Type
        - Return to Menu
```

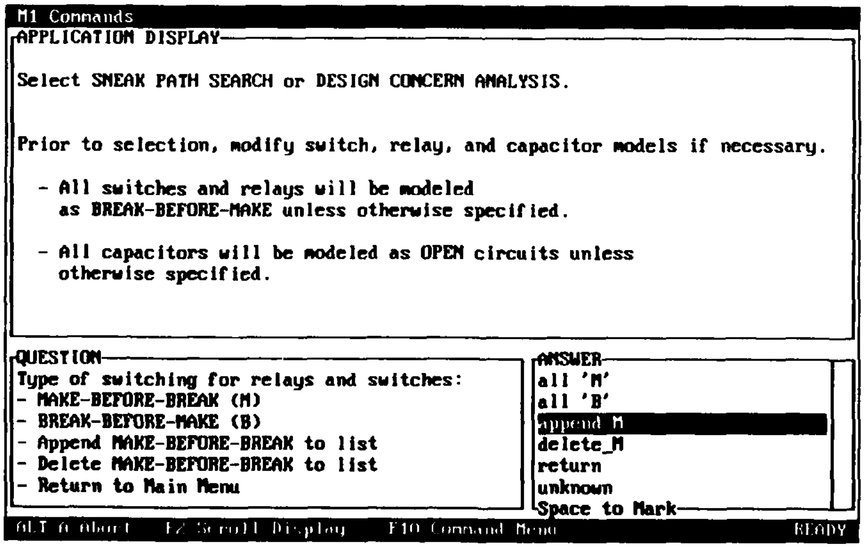
```
continue
power_list
gnd_list
circuit_type
return
unknown
```

Figure A-18. Design Parameter Summary



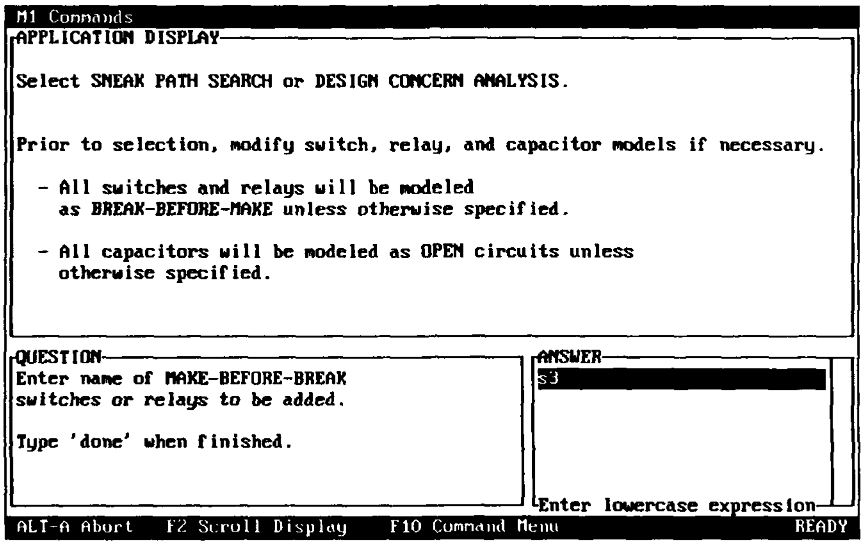
```
M1 Commands
APPLICATION DISPLAY

  The following topics are covered under FUNCTIONAL guidelines:
    -- Power distribution and power switching
    -- Digital timing
    -- status indicators (i.e. lamps, LEDs)

  The following topics are covered under DEVICE guidelines:
    -- Bipolar transistors
    -- Op-amps
    -- Noise sensitive circuits (op-amps, one-shots, SCRs)
    -- Relays
    -- TTL or MOS digital circuitry

QUESTION                                        ANSWER
Choose desired operation:                       return
                                                functional_guideln
  - return to main menu                         device_guideln
  - functional guideline                        unknown
  - device guideline

                                                Space to Mark
ALT-A Abort   F2 Scroll Display   F10 Command Menu              READY
```

Figure A-19. Design Concern Type Menu



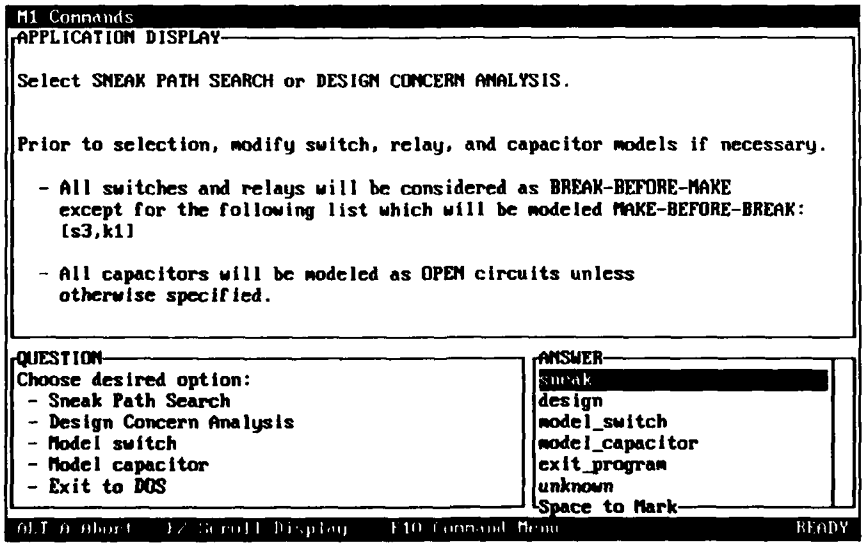
```
M1 Commands
APPLICATION DISPLAY

Select SNEAK PATH SEARCH or DESIGN CONCERN ANALYSIS.


Prior to selection, modify switch, relay, and capacitor models if necessary.

  - All switches and relays will be modeled
    as BREAK-BEFORE-MAKE unless otherwise specified.

  - All capacitors will be modeled as OPEN circuits unless
    otherwise specified.

QUESTION                                        ANSWER
Type of switching for relays and switches:     all 'M'
- MAKE-BEFORE-BREAK (M)                         all 'B'
- BREAK-BEFORE-MAKE (B)                         append M
- Append MAKE-BEFORE-BREAK to list              delete_M
- Delete MAKE-BEFORE-BREAK to list              return
- Return to Main Menu                           unknown
                                                Space to Mark
ALT A Abort   F2 Scroll Display   F10 Command Menu              READY
```

Figure A-20. Switch/Relay Model Menu



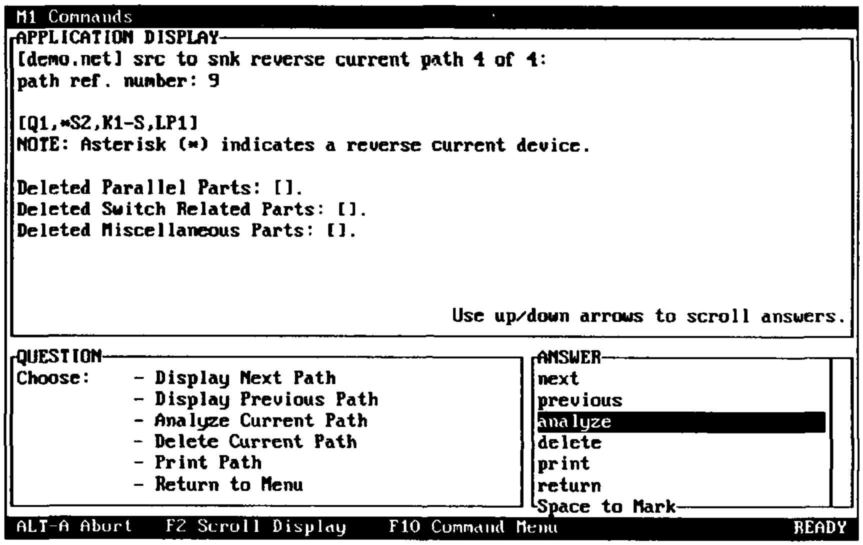
```
M1 Commands
APPLICATION DISPLAY

Select SNEAK PATH SEARCH or DESIGN CONCERN ANALYSIS.


Prior to selection, modify switch, relay, and capacitor models if necessary.

  - All switches and relays will be modeled
    as BREAK-BEFORE-MAKE unless otherwise specified.

  - All capacitors will be modeled as OPEN circuits unless
    otherwise specified.

QUESTION                                           ANSWER
Enter name of MAKE-BEFORE-BREAK                    s3
switches or relays to be added.

Type 'done' when finished.

                                                   Enter lowercase expression
ALT-A Abort   F2 Scroll Display   F10 Command Menu                      READY
```

Figure A-21. Switch Configuration Menu



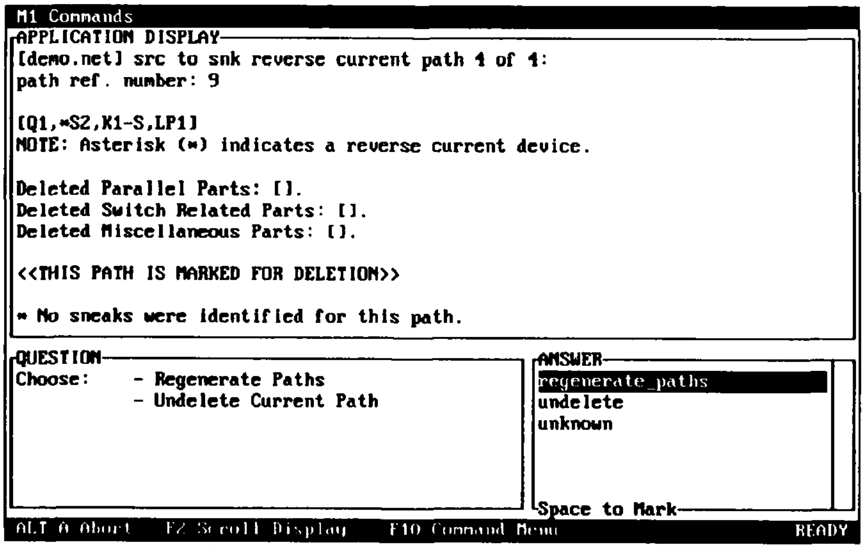
```
M1 Commands
APPLICATION DISPLAY

Select SNEAK PATH SEARCH or DESIGN CONCERN ANALYSIS.


Prior to selection, modify switch, relay, and capacitor models if necessary.

  - All switches and relays will be considered as BREAK-BEFORE-MAKE
    except for the following list which will be modeled MAKE-BEFORE-BREAK:
    [s3,k1]

  - All capacitors will be modeled as OPEN circuits unless
    otherwise specified.

QUESTION                                           ANSWER
Choose desired option:                             sneak
 - Sneak Path Search                               design
 - Design Concern Analysis                         model_switch
 - Model switch                                    model_capacitor
 - Model capacitor                                 exit_program
 - Exit to DOS                                     unknown
                                                   Space to Mark
ALT-A Abort   F2 Scroll Display   F10 Command Menu                      READY
```

Figure A-22. Main Menu Listing M-B-B Switches



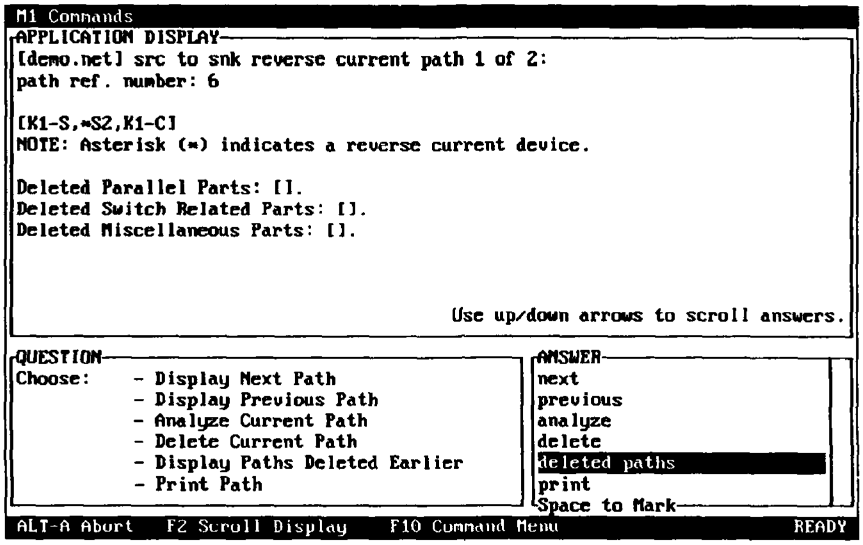
```
M1 Commands
APPLICATION DISPLAY
[demo.net] src to snk reverse current path 4 of 4:
path ref. number: 9

[Q1,*S2,K1-S,LP1]
NOTE: Asterisk (*) indicates a reverse current device.

Deleted Parallel Parts: [].
Deleted Switch Related Parts: [].
Deleted Miscellaneous Parts: [].



                                         Use up/down arrows to scroll answers.
QUESTION                                                 ANSWER
Choose:     - Display Next Path                          next
            - Display Previous Path                      previous
            - Analyze Current Path                       analyze
            - Delete Current Path                        delete
            - Print Path                                 print
            - Return to Menu                             return
                                                         Space to Mark
ALT-A Abort    F2 Scroll Display    F10 Command Menu                      READY
```

Figure A-23. Reverse Current Path 4



```
M1 Commands
APPLICATION DISPLAY
[demo.net] src to snk reverse current path 4 of 4:
path ref. number: 9

[Q1,*S2,K1-S,LP1]
NOTE: Asterisk (*) indicates a reverse current device.

Deleted Parallel Parts: [].
Deleted Switch Related Parts: [].
Deleted Miscellaneous Parts: [].

<<THIS PATH IS MARKED FOR DELETION>>

* No sneaks were identified for this path.
QUESTION                                                 ANSWER
Choose:     - Regenerate Paths                           regenerate_paths
            - Undelete Current Path                      undelete
                                                         unknown



                                                         Space to Mark
ALT-A Abort    F2 Scroll Display    F10 Command Menu                      READY
```

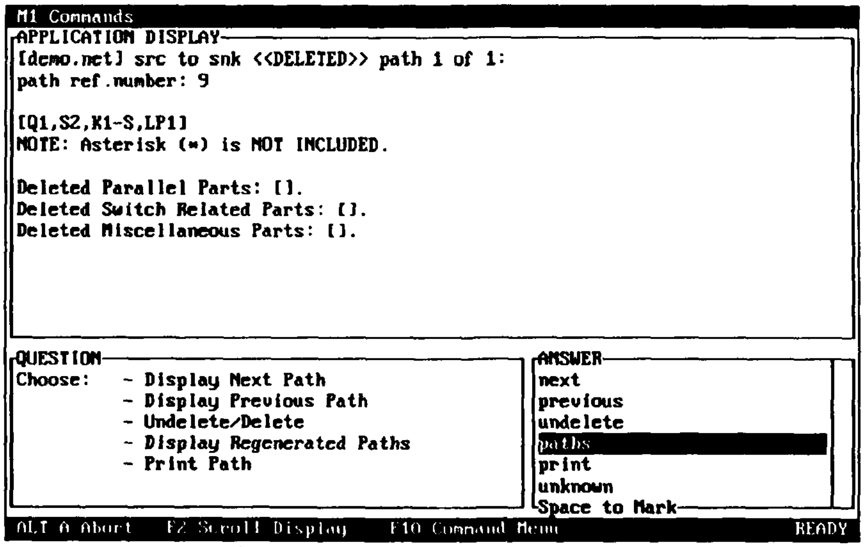
Figure A-24. Path Marked for Deletion



```
[demo.net] src to snk reverse current path 1 of 2:
path ref. number: 6

[K1-S,*S2,K1-C]
NOTE: Asterisk (*) indicates a reverse current device.

Deleted Parallel Parts: [].
Deleted Switch Related Parts: [].
Deleted Miscellaneous Parts: [].


                                    Use up/down arrows to scroll answers.
```

```
Choose:    - Display Next Path
           - Display Previous Path
           - Analyze Current Path
           - Delete Current Path
           - Display Paths Deleted Earlier
           - Print Path
```

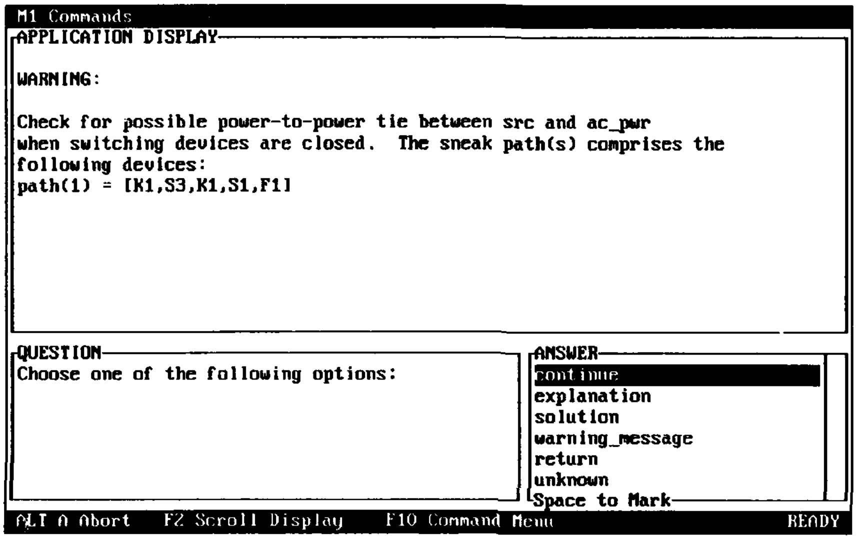
Figure A-25. Regenerated Reverse Current Path 1



```
[demo.net] src to snk <<DELETED>> path 1 of 1:
path ref.number: 9

[Q1,S2,K1-S,LP1]
NOTE: Asterisk (*) is NOT INCLUDED.

Deleted Parallel Parts: [].
Deleted Switch Related Parts: [].
Deleted Miscellaneous Parts: [].
```

```
Choose:   - Display Next Path
          - Display Previous Path
          - Undelete/Delete
          - Display Regenerated Paths
          - Print Path
```

Figure A-26. Deleted Paths Screen



```
APPLICATION DISPLAY

WARNING:

Check for possible power-to-power tie between src and ac_pwr
when switching devices are closed.  The sneak path(s) comprises the
following devices:
path(1) = [K1,S3,K1,S1,F1]
```

```
QUESTION
Choose one of the following options:
```

```
ANSWER
continue
explanation
solution
warning_message
return
unknown
Space to Mark
```

Figure A-27. Power-to-Power Tie Warning Message

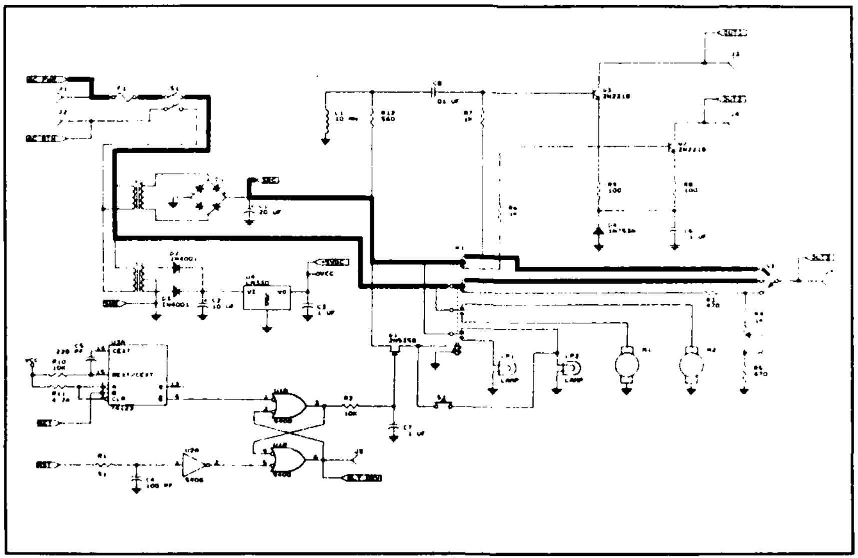


Figure A-28. Power-to-Power Tie Highlighted on Schematic

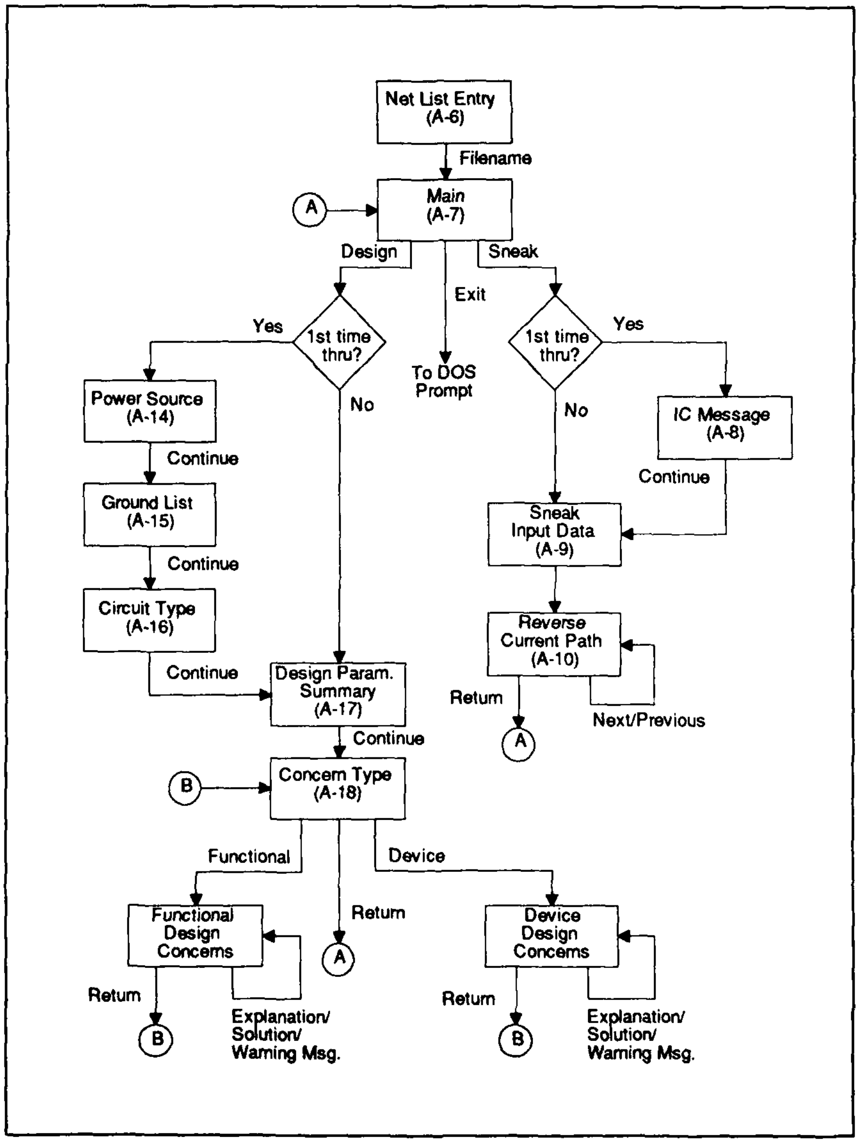


Figure A-29. SCAT Program Flow

APPENDIX B

SNEAKS

The following instructions are provided for applying the Design Rules and Guidelines appearing in sections B.1, B.2, and B.3. These aids primarily address power and ground distribution to analog and digital circuitry although some guidelines also address signal distribution. The aids are intended to be integrated with the circuit design effort early in the development process. Therefore, it is expected that the aids will be applied to a functional diagram or circuit schematic at the assembly (e.g. PC board) or subsystem (e.g. power distribution) level. The aids are applied to additional circuitry as the design progresses. For each application, it is important to account for all present or anticipated interfaces with other circuitry, particularly connections to sources of power and ground. If specific circuit paths are not defined at the time the aids are applied, represent these interfaces functionally and repeat the analysis when the interface definition becomes more refined. Specific instructions are as follows:

(1) During the early design phase follow the Sneak Circuit Design Rules found in section B.1. The rules are intended for the circuit designer during the concept and validation development phases. Adherence to these rules will avoid many common causes of sneak circuits.

(2) Apply the Sneak Circuit Functional Guidelines (section B.2) when functional diagrams are available at the subsystem level. The diagrams should depict power and ground distribution paths and power switching elements. The remaining functions can be depicted as power supply loads. For complex systems, focus the application of the rules on the circuitry associated with critical system functions, for example those affecting loss of life or destruction of the system, may be identified by fault tree analysis or funtional FMEA.

(3) Apply the Sneak Circuit Device Guidelines (section B.3) when the design has progressed to where circuit schematics are available at the assembly (i.e. circuit board) level. The schematics must include all power and signal paths, including connections across subsystem interfaces. Again, the extent of the analysis can be reduced by limiting it to the critical areas of the system.

B.1 SNEAK CIRCUIT DESIGN RULES

The Sneak Circuit Design Rules guide the circuit designer to avoid networks commonly associated with sneak conditions. Each rule is formatted as follows:

a. PROBLEM -- A statement of the sneak problem addressed by the rule.

b. SOLUTION -- A recommended approach for implementing the rule in practical situations.

c. Figures depicting circuitry violating and complying with the rule.

d. Supplementary information further explaining the rule.

The following is a selection guide for application of the design rules:

RULE	APPLICABLE CIRCUITRY	PAGE
1	Multiple Power Sources and Returns	B-3
2	Ground Side Switching	B-4
3	Circuit Symmetry	B-5
4	Power and Ground Connectors	B-6
5	Wired-OR Circuits	B-7
6	Sneak Timing in Memory Power Switching	B-8
7	Switch Labeling	B-9

Rule 1. MULTIPLE POWER SOURCES AND RETURNS

PROBLEM: Sneak paths involving multiple power sources and/or multiple ground returns.

SOLUTION: Structure circuits so that all current for a given load flows from one power source to one ground return. Where this is not possible, isolate power sources using diodes for DC power or relays (electromechanical or solid-state) for AC or DC power. Use Schottky diodes or relays for DC applications requiring very low voltage drop and power dissipation. Isolate returns by separating high and low current loads.

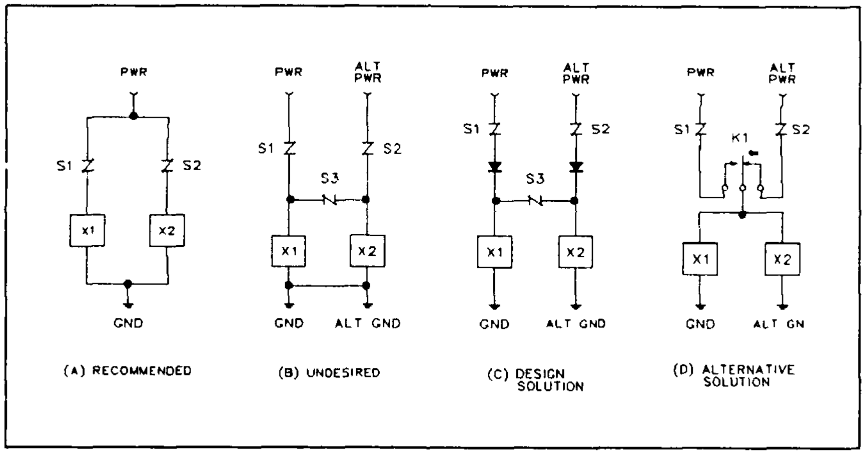


Figure B-1. MULTIPLE POWER SOURCES AND RETURNS

Adherence to this rule avoids "Y," "X" and "H" circuit patterns associated with multiple power sources and sinks. This is a general rule to be followed wherever possible. An example of a network complying with this rule appears in Figure B-1(A), and an example of a network violating it appears in part (B) of the figure. The violations shown can result in power-to-power or ground-to-ground ties. Isolation must be provided to avoid the mixing of low current and high current ground returns. Examples are shown in parts (C) and (D) of the figure.

Rule 2. GROUND SIDE SWITCHING

PROBLEM: **Sneak paths caused by interrupting current at the ground side of a load.**

SOLUTION: **Do not place current interrupting elements (e.g. switches, relays, circuit breakers, fuses) in ground return paths. If connectors are required, apply Rule 3.**

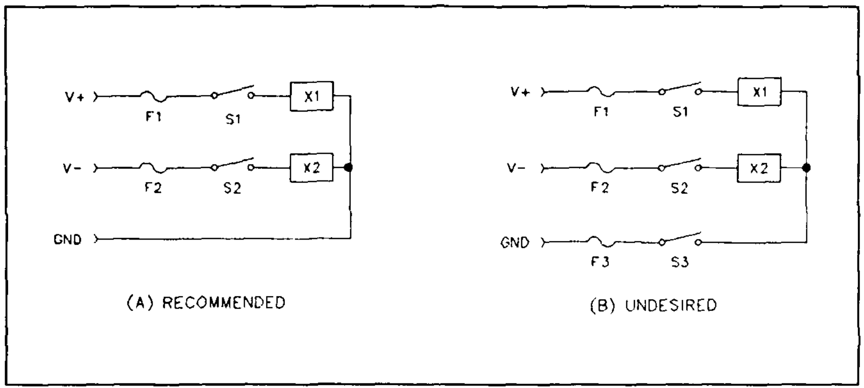


Figure B-2. GROUND-SIDE SWITCHING

An arbitrary number of current interrupters (switches, relays, connectors, fuses, circuit breakers, etc.) can be placed on the supply side of the load without danger of causing reverse current flow. However, placement of switches on the ground side of the load can under some conditions cause a sneak circuit. The placement of switches on the ground side of loads is an undesirable practice (it is prohibited in most circumstances in the National Electrical Code), but connectors on the ground side may be necessary and can in principle cause the same sneak circuit.

An example of problems caused by ground-side switching is shown in Figure B-2. In part (A), loads X1 and X2 are powered respectively from positive and negative DC voltage supplies. In part (B), if fuse F3 opens before F1 and F2 or if pole S3 opens before S1 and S2, a voltage equal to sum of V+ and V- will be distributed across the loads according to the ratio of their impedances for the duration of this condition.

Rule 3. CIRCUIT SYMMETRY

PROBLEM: **Sneak paths caused by connectors (or other current interrupting devices) located on the ground side of a load.**

SOLUTION: **When placement of a connector at the ground side of the load is required, keep the supply and ground return paths symmetrical.**

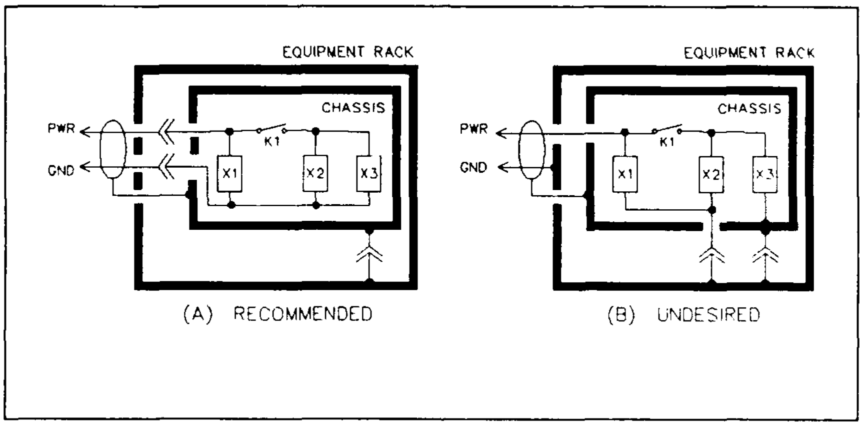


Figure B-3. CIRCUIT SYMMETRY AND GROUND-SIDE CONNECTORS

The switching topology from the supply to a load should be duplicated from the load to ground. A "duplicated topology" implies identical branching but not necessarily the same number or type of switching elements per branch. Adhering to this rule will avoid inadvertent, topological H-patterns that are a common source of reverse current flow. A practical equipment configuration is shown in Figure B-3. In part (A) of that figure all ground return paths share the same connector with the power source line. In part (B) loads X1 and X2 are directly connected to the equipment rack ground and the return for X3 is connected to the chassis ground to which the cable shield is also attached. The sneak path occurs when the equipment is powered up with contactor K1 open, and the chassis removed from the rack. The latter condition is frequently encountered during troubleshooting or when performing depot level maintenance. Sneak current flows from the power input through X1, through X2 in the reverse direction, through X3 and returns through the power cable shield. If the loads represent a low impedance, the current through the shield can reach sufficiently high values to cause excessive heat and in extreme cases fire. An important area of concern is possible latent damage due to the reverse current flow through X2 and the heating of the wire(s) surrounded by the shield.

Rule 4. POWER AND GROUND CONNECTORS

PROBLEM: Sneak paths caused by power and ground connectors.

SOLUTION: Avoid the use of separate connectors for providing power and ground return lines to a circuit.

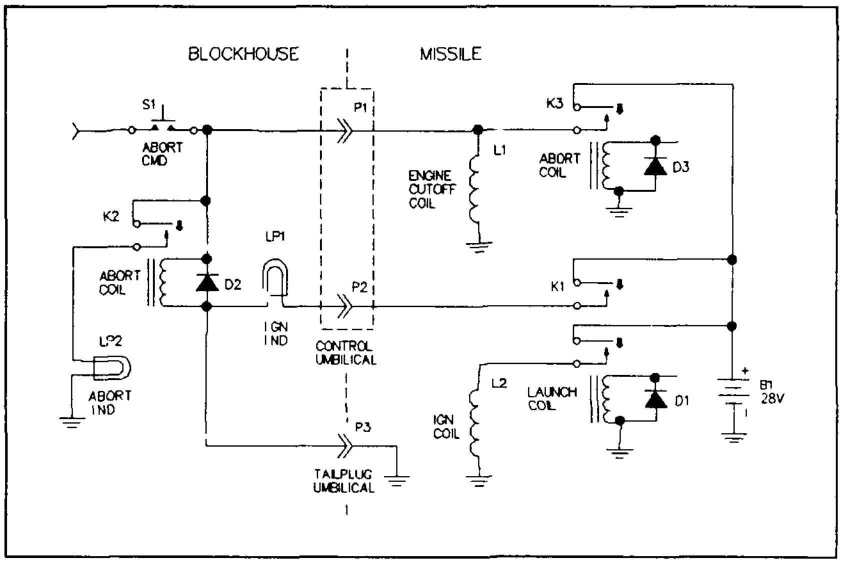


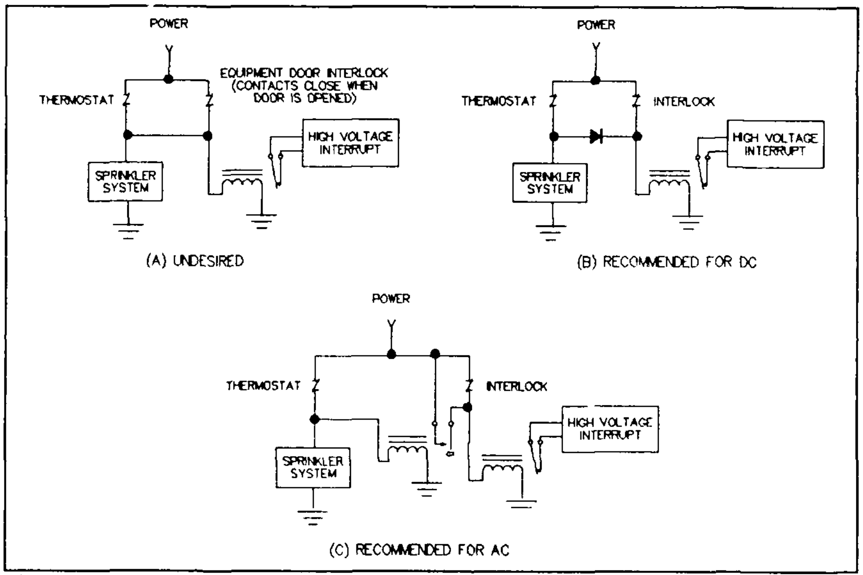
Figure B-4. UNDESIRABLE CONSEQUENCES OF SEPARATE POWER AND GROUND CONNECTORS

The design solution for this rule occurs as a consequence of adhering to the symmetry rule (Rule 3) but is important enough to merit special consideration. Adherence to this simple rule would have prevented the Mercury-Redstone launch failure (as described on page 5, para 2). This is illustrated in Figure B-4 which depicts the portion of the blockhouse and missile circuitry involving the sneak. As shown in the figure, the normal sequence of events is for launch command relay K1 to be triggered, enabling battery B1 to power the missile ignition coil and the blockhouse ignition indicator. However, if upon launch the tail plug umbilical carrying ground return P3 separated before the control umbilical, current flowing through the ignition indicator would continue through diode D2 and supply power to the engine cutoff coil. This sequence of events actually did occur. It could have been prevented by routing the power and return lines through the same connector.

Rule 5. WIRED-OR CIRCUITS

PROBLEM: Sneak paths caused by selecting alternate paths in "wired-OR" circuits.

SOLUTION: The "wired-OR" can be used only where the effect to be produced by the alternate conditions is <u>exactly</u> the same. When some conditions are intended to cause additional or modified effects, isolation must be provided.



(A) UNDESIRED

(B) RECOMMENDED FOR DC

(C) RECOMMENDED FOR AC

Figure B-5. **WIRED-OR INTERLOCK**

This problem is frequently encountered in alarm and interlock circuits. A simple example is that of an interlock actuated by two or more conditions, as shown in Figure B-5. A high voltage power supply is to be interrupted when either the equipment door is opened or when the ambient temperature exceeds a preset level. The temperature sensor is also intended to operate an automatic sprinkler system. In the implementation shown in Figure B-5(A), opening the equipment door will interrupt the high voltage as intended, but it will also unintentionally turn on the sprinkler system. If the protective circuit operates on DC the sneak path can be eliminated by the addition of a diode in the horizontal branch as shown in part (B) of the figure. If the circuit utilizes AC, or if the diode is not desired for other reasons, a relay having a normally open contact arrangement is added as shown in part (C) of the figure.

Rule 6. SNEAK TIMING IN MEMORY POWER SWITCHING

PROBLEM: Sneak timing due to momentary loss of power to volatile computer memory and other essential loads during switch-over to an alternate power source.

SOLUTION: For small memories, use break-before-make switching and sufficient capacitance to maintain the voltage during switch-over. For large memories or to protect against a short on the main supply, use make-before-break switching and diode isolation.

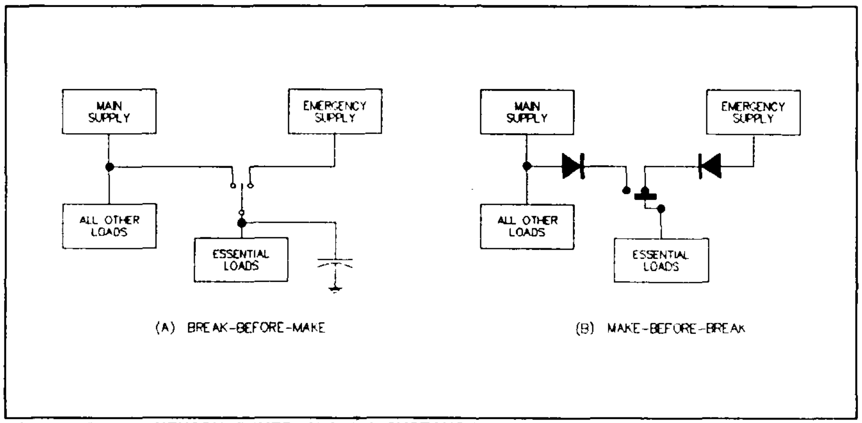


Figure B-6. MEMORY POWER SUPPLY SWITCHING

This rule involves inappropriate selection of a break-before-make or make-before-break switch contact arrangement. As shown in Figure B-6, an essential load is normally connected to the main supply but can be switched to an auxiliary supply when the main voltage drops below a set threshold. The implementation of Figure B-6(A) uses break-before-make contacts and will subject the essential load to a brief power interruption. This interruption can cause complete or partial loss of all data stored in a volatile computer memory. If a computer is included in the sensitive load, protection against data loss must be provided. Where the computer memory is comparatively small, a capacitor across the memory power supply (Figure B-6(A)) may be sufficient to maintain the voltage at a safe level during the switching interval when the break-before-make configuration is used. If this will not be adequate, the make-before-break switch must be used (Figure B-6(B)) with further isolation of the non-essential load (either by diode or fast-acting switch) to protect against the case where the drop in the main voltage was due to a short circuit in the main supply or in the non-volatile load.

Rule 7. SWITCH LABELING

PROBLEM: **Sneak label causing an action opposite to the one intended to occur when a switch is toggled.**

SOLUTION: **Label switches according to the action performed in addition to the object being controlled.**

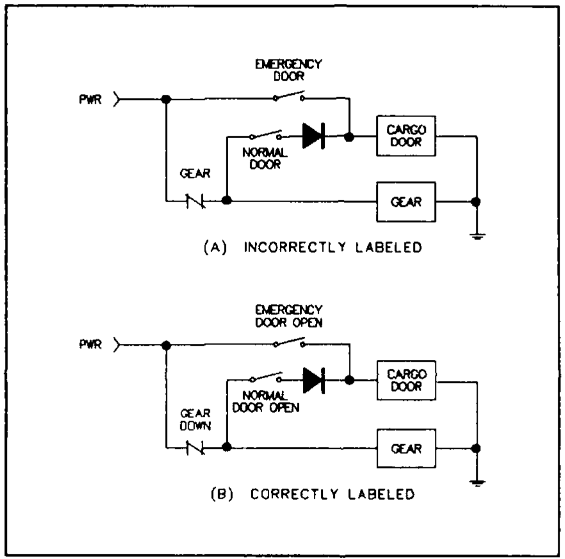


Figure B-7. SWITCH LABELING

In Figure B-7(A), the labels EMERGENCY DOOR, NORMAL DOOR and GEAR are ambiguous as to whether closure of their corresponding switches occurs for "door open" or "door closed" or for "gear up" or "gear down". This ambiguity is eliminated by the labels shown in Figure B-7(B).

B.2 SNEAK CIRCUIT FUNCTIONAL GUIDELINES

The Sneak Circuit Functional Guidelines aid the circuit designer or reliability analyst to identify functional networks commonly associated with sneak conditions. Each guideline is formatted as follows:

a. TARGET -- The types of circuit functions targeted by the guideline.

b. PROBLEM -- A statement of the sneak problem addressed by the rule.

c. SOLUTION -- A recommended approach for implementing the rule in practical situations.

d. COMMENT -- Supplementary information further explaining the rule, in some cases accompanied by a figure.

The following is a selection guide for application of the functional guidelines:

APPLICABLE CIRCUITRY	PAGE
Power Distribution Circuits	B-11
Switching Circuitry	B-18
Sneak Timing	B-20
Sneak Labels and Indications	B-23

POWER DISTRIBUTION CIRCUITS

TARGET: Primary and secondary power distribution circuitry comprising power sources, ground returns, switches, contactors, relays, circuit breakers, fuses, solid state switches, connectors.

PROBLEM: Asymmetrical pattern of connections for power distribution and ground return circuitry.

SOLUTION: Use the same circuit connection topology for the supply side and ground side of a load. Use the same connector for symmetrical power and ground connections.

COMMENT: Circuit connection symmetry for power and ground distribution implies an identical number and location of power and ground connections feeding a load. Asymmetrical connections can cause sneak paths as shown in Figure B-8. In part (A) of the figure, power connection J3 has no counterpart on the ground side of load X2. If connections J2 and J3 are open while the remainder are closed, current can unintentionally flow in the reverse direction through X2. This problem has been eliminated in part (B) of the figure by the inclusion of connection J3-2.

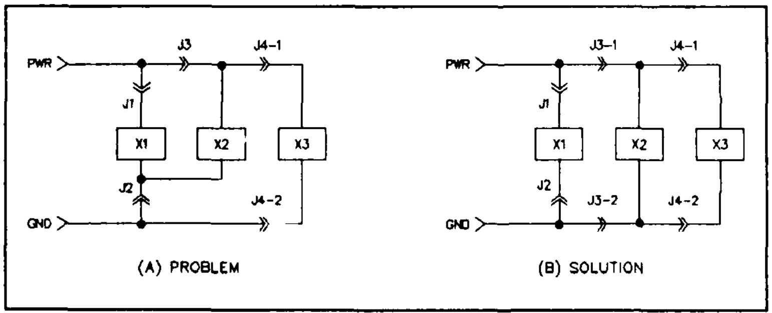


Figure B-8. SYMMETRICAL POWER DISTRIBUTION

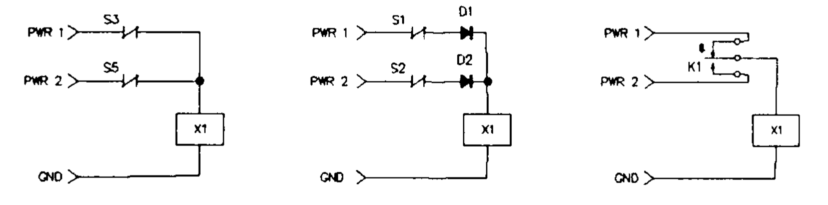
POWER DISTRIBUTION CIRCUITS (Continued)

TARGET: Primary and secondary power distribution circuitry comprising power sources, ground returns, switches, contactors, relays, circuit breakers, fuses, solid state switches, connectors.

PROBLEM: Power-to-power tie between supplies providing power to a common load.

SOLUTION: For DC power, add diodes to isolate the supplies. For AC power or as an alternative for DC power, use a double-throw relay or switch having a break-before-make contact arrangement to select either supply, and provide adequate capacitance at the load to hold up the supply voltage during switch-over.

COMMENT: Referring to part (A) of Figure B-9, PWR 1 and PWR 2 will be tied if switches S1 and S2 are simultaneously engaged. A make-before-break switch contact arrangement can cause a momentary power tie of this type. Part (B) shows the addition of isolation diodes. Schottky diodes can be used to minimize the diode voltage drop. Part (C) depicts the use of a single-pole, double-throw relay.



(A) PROBLEM

(B) SOLUTION

(C) ALTERNATE SOLUTION

Figure B-9. PREVENTING A POWER-TO-POWER TIE

POWER DISTRIBUTION CIRCUITS (Continued)

TARGET: Primary and secondary power distribution circuitry comprising power sources, ground returns, switches, contactors, relays, circuit breakers, fuses, solid state switches, connectors.

PROBLEM: Multiple supplies unintentionally enabling a shared load.

SOLUTION: **Analyze the logic and timing of the power control circuitry to insure that all power sources sharing a common load are switched off when the load must be disabled.**

COMMENT: This problem can occur when supplies are OR'd (tied through diodes) as depicted earlier in Figure B-9 or when supplies power a common system as depicted in Figure B-11. In the latter, a low voltage power supply (LVPS) and a high voltage power supply (HVPS) provide DC power to a CRT display system. In part (A) of the figure, circuitry is provided to shutdown the LVPS in the event of an over-voltage or over-current condition at the power supply output. However, the protection circuitry does not shut down the HVPS. This can result in possible damage to the CRT screen (as the LVPS shuts down, the screen can be burned if the CRT deflection drives collapse before the video drive) and can also present a potential maintenance hazard (the absence of low voltage power may lead one to erroneously assume that all power is shut off).

Solutions to this problem are shown in parts (B) and (C) of Figure B-11. In part (B), the HVPS is provided with a fast shutdown function activated by the shutdown signal from the LVPS. In part (C), the HVPS is replaced with one that is powered from the low voltage DC generated by the LVPS. As the LVPS shuts down, so does the HVPS. The actual implementation of this solution requires further analysis to determine if the HVPS will shut down fast enough to prevent screen burn.

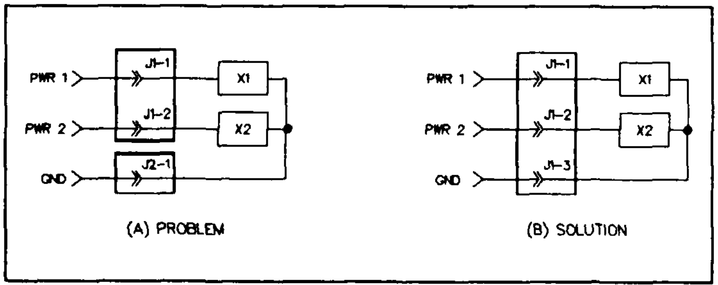
POWER DISTRIBUTION CIRCUITS (Continued)

TARGET: Primary and secondary power distribution circuitry comprising power sources, ground returns, switches, contactors, relays, circuit breakers, fuses, solid state switches, connectors.

PROBLEM: Power-to-power path between supplies providing power to loads sharing a common, independently switchable ground.

SOLUTION: Do not place switching elements other than those associated with circuit connections on the ground side of a load. For circuit connections, combine power and ground connections in the same connector.

COMMENT: The problem is two-fold. As can be seen from part (A) of Figure B-10 below, loss of ground causes (1) a reverse current to flow through the load connected to the lower of the two supplies, and (2) the power-to-power voltage will divide according to the ratio of the load impedances, and the resulting voltage will appear on the ground side of the loads, presenting a potential safety hazard to personnel. The solution follows as a consequence of adhering to the general circuit symmetry guideline depicted in Figure B-8. In part (B) of Figure B-10, power and ground connections are combined in the same connector, thereby preventing loss of ground without loss of power.



(A) PROBLEM

(B) SOLUTION

Figure B-10. PREVENTING THE INDEPENDENT LOSS OF GROUND

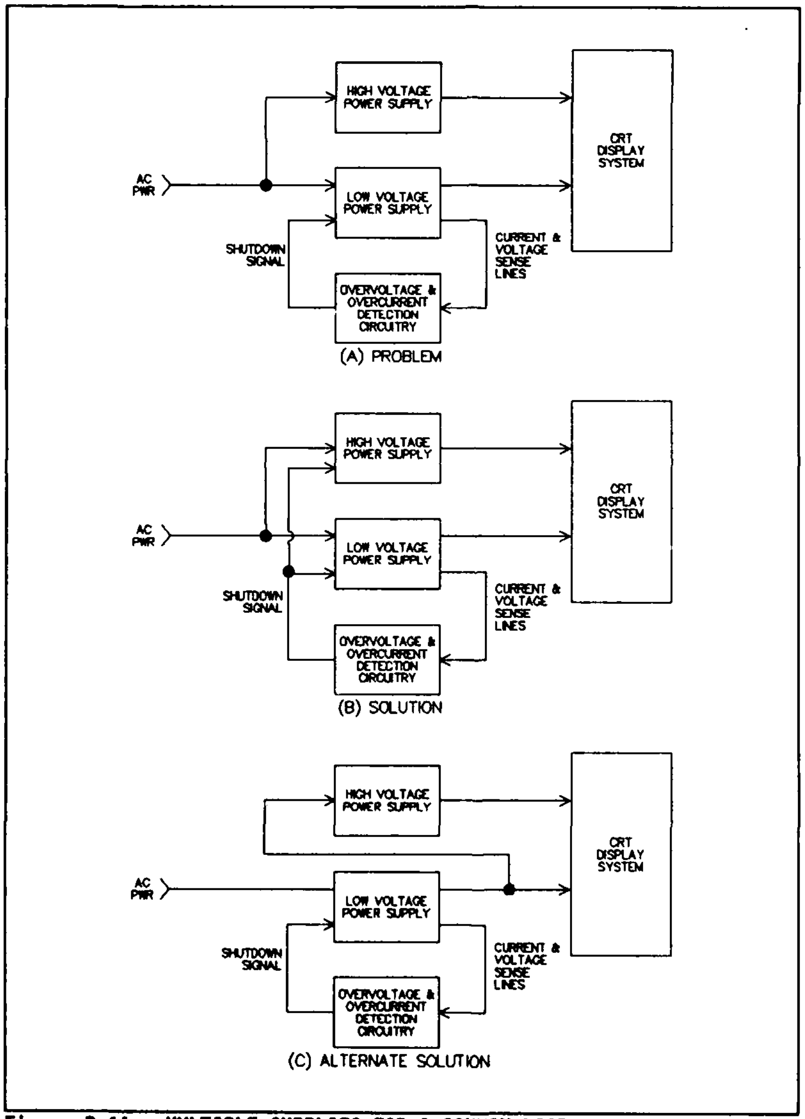

HIGH VOLTAGE POWER SUPPLY
CRT DISPLAY SYSTEM
AC PWR
LOW VOLTAGE POWER SUPPLY
SHUTDOWN SIGNAL
CURRENT & VOLTAGE SENSE LINES
OVERVOLTAGE & OVERCURRENT DETECTION CIRCUITRY
(A) PROBLEM
HIGH VOLTAGE POWER SUPPLY
CRT DISPLAY SYSTEM
AC PWR
LOW VOLTAGE POWER SUPPLY
SHUTDOWN SIGNAL
CURRENT & VOLTAGE SENSE LINES
OVERVOLTAGE & OVERCURRENT DETECTION CIRCUITRY
(B) SOLUTION
HIGH VOLTAGE POWER SUPPLY
CRT DISPLAY SYSTEM
AC PWR
LOW VOLTAGE POWER SUPPLY
SHUTDOWN SIGNAL
CURRENT & VOLTAGE SENSE LINES
OVERVOLTAGE & OVERCURRENT DETECTION CIRCUITRY
(C) ALTERNATE SOLUTION

Figure B-11. MULTIPLE SUPPLIES FOR A COMMON LOAD

PROBLEM: **A difference in ground potential between two interfacing assemblies (e.g. two PC boards).**

SOLUTION: **Insure interfacing circuitry share a common ground. Minimize voltage difference caused by IR drop by keeping ground return paths between assemblies as short as possible and by using adequate wire gauges or bus bars.**

COMMENT: Ground voltage differences between interfacing circuitry can cause IC input substrate diodes to become heavily forward biased, thereby damaging the device. This problem is shown in Figure B-12. In part (A) of the figure, a logic gate drives a similar gate located 10 feet away. Diode D1 represents the chip substrate diode in the receiving gate. The receiving gate obtains its ground from the driver gate circuit. One amp of current flows through this ground return, and the wire gauge is 28 AWG or approximately 70 ohms per 1000 feet. The resulting ground potential difference (v_2 - v_1 in the figure) is 0.7 volts excluding transient voltages arising from the line inductance. This voltage difference is enough to forward bias D1 when the driver outnut is in its low state. In part (B) of the figure, the wire gauge has been increased to 18 AWG (approximately 3 times the overall diameter of the 28 AWG wire). This lowers the wire resistance (and therefore the ground potential difference) by a factor of 10. In addition, the gates have been replaced with transmission line driver and receiver devices that tolerate negative signals. Ground voltage differences also distort signal amplification in analog circuitry and lower input noise margin in digital circuitry. The layout of ground circuitry should anticipate the eventual increases in resistance of the ground path such as due to corrosion and electromigration. The selection of materials and layout should minimize these increases.

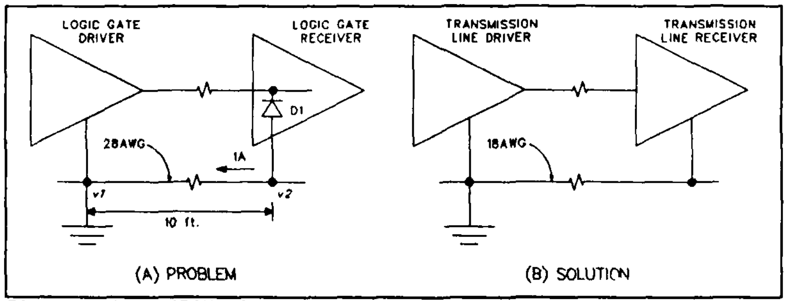
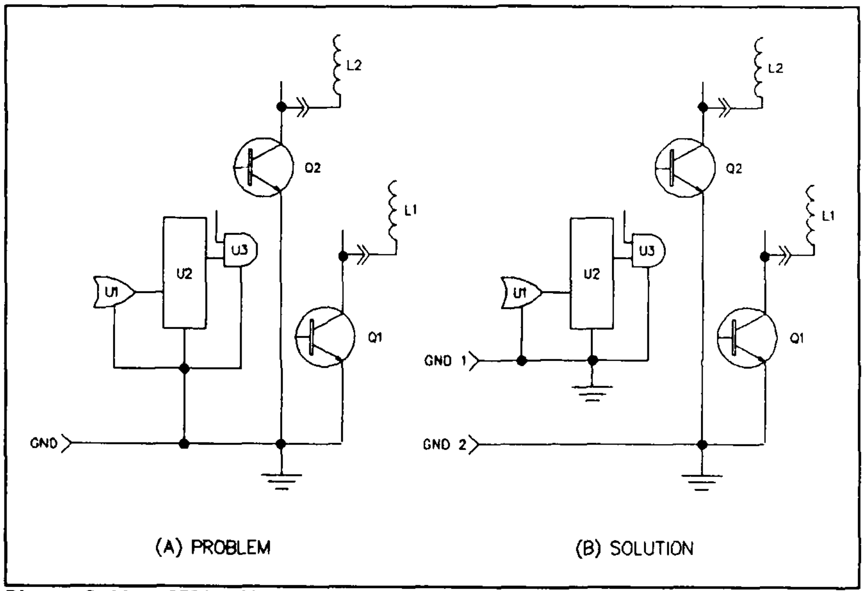


Figure B-12. **GROUNDING**

PROBLEM: Mixing high current and low current grounds within a circuit.

SOLUTION: Provide separate ground return paths for high current and low current loads.

COMMENT: High current loads include drivers for displays, motor windings, relay coils. Low current loads include logic and low power analog circuitry. Separating the grounds for these two types of loads prevents voltage transients due to the resistance and inductance of the high current path from being introduced into the low current circuitry. This is depicted in Figure B-13 where L1 and L2 represent high current inductive loads, Q1 and Q2 are the load drivers, and U1, U2 and U3 represent miscellaneous low current logic devices. In part (A) of the figure, a single ground is used for the high current drivers and the low current devices. In part (B), separate grounds originating at the power supply are used. Separating analog from digital grounds and low frequency from high frequency grounds is also a good design practice.



(A) PROBLEM

(B) SOLUTION

Figure B-13. SEPARATING HIGH AND LOW CURRENT GROUNDS

SWITCHING CIRCUITRY

TARGET: Switching devices controlling shared loads.

PROBLEM: A load connected to more than one switch and unintentionally enabled by a switch being closed.

SOLUTION: Place a diode or relay between the load and the switch in question.

COMMENT: This problem is typically associated with paralleled switches connected to paralleled loads in an "X" pattern as shown in part (A) of Figure B-14, and leads to the wired-OR problem presented in Design Rule 5. The solution is to replace the "X" with an "H" using a diode (D3 in part (B) of the figure) for the cross-bar.

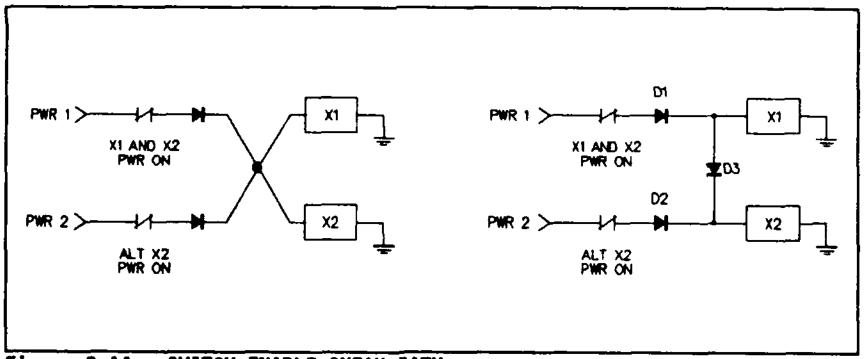


Figure B-14. SWITCH ENABLE SNEAK PATH

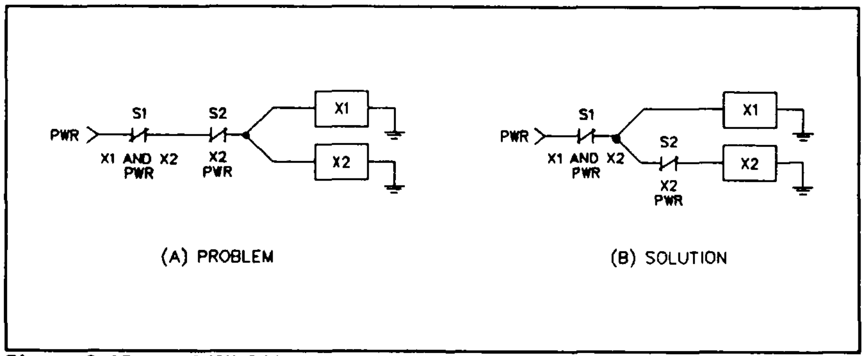
SWITCHING CIRCUITRY (Continued)

TARGET: Switching devices controlling shared loads.

PROBLEM: A load controlled by more than one switch and unintentionally disabled by a switch being opened.

SOLUTION: Connect the load in question to a switch directly tied to the power source, or add a switch dedicated to controlling the load in question.

COMMENT: This problem is associated with switches in series connected to paralleled loads in a "Y" pattern, the stem being the switch string (S1 and S2 in part (A) of Figure B-15). It is most desireable for the load in question to be controlled by the switch directly tied to the power source (switch S1 in the figure) so that the load in question can be connected directly to that switch instead of being directly connected in parallel with the remaining loads.



(A) PROBLEM

(B) SOLUTION

Figure B-15. SWITCH DISABLE SNEAK PATH

SNEAK TIMING

TARGET: Digital circuitry.

PROBLEM: Logic and timing errors caused by a digital signal that splits and later recombines.

SOLUTION: Analyze the signal path through a complete cycle (<u>e.g.</u> ON-OFF-ON) to insure correct logic and timing. Correct timing skew problems by providing a clocked data buffer (e.g. latch) to sample stable data at the point where they recombine.

COMMENT: Recombined paths often lead to sneak timing as a result of the logic functions performed along each path. A more commonly encountered problem is a transient signal ("glitch") caused by differences in the signal propagation delay between paths. A clocked buffer reduces timing offset ("skew") by resynchronizing the signals. For example, in part (A) of Figure B-16 a glitch occurs at the output of gate U4 during the brief interval that the skewed output signals at gates U2 and U3 are both high. The glitch is prevented as shown in part (B) of the figure by sampling the outputs of U2 and U3 with buffer U5 after the signals have completed their transitions.

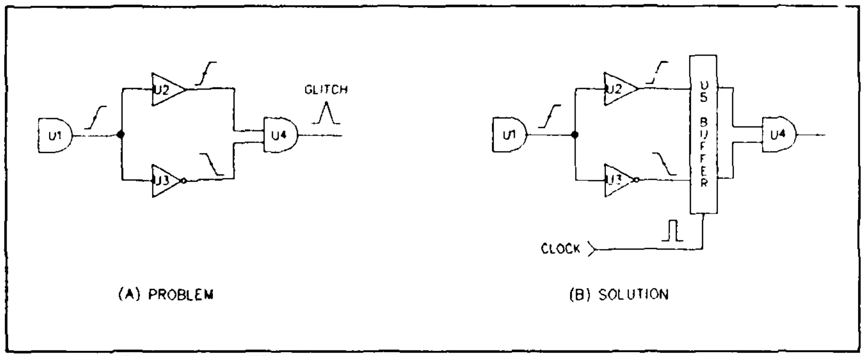


Figure B-16. RECOMBINING DIGITAL SIGNALS

SNEAK TIMING (Continued)

TARGET: Digital circuitry.

PROBLEM: False data caused by interfacing digital devices powered from different supplies.

SOLUTION: Insure interconnected digital devices share a common power supply. If this is not possible, circuit outputs must be considered invalid until all supplies are powered up and all registers reset to their initial states.

COMMENT: When the supply voltage is below some threshold value during power up or power down, the output of a digital device is unpredictable. During power down, a subsequent device powered from a supply which decays more slowly may therefore receive unpredictable data and produce false data. Similarly, during power up, a subsequent device powered from a supply which rises faster may create the same problem. As shown in Figure B-17, this type of problem is likely to occur at an interface. Part (A) of the figure depicts an example of power-on and power-off waveforms for the +5 volt and +12 volt power supplies used in the circuit shown in part (B). At time t_1 the +12 volt power has reached its operating value while the +5 volts is still below a level at which predictable operation of the logic devices is guaranteed (e.g. 4.5 volts for TTL logic families). In part (C), a circuit has been added to inhibit gate U2 until the +5 volts has reached an operating level. The time delay for this circuit is set by the product of resistor R1 and capacitor C1 and the threshold of Schmitt triggered gate U4. A gate having a Schmitt triggered input is required to unambiguously sense the slowly rising voltage across C1 (see for example Figure B-22 and its accompanying guideline). The inhibit function is not required at power-off for this example because the +12 volts powering U3 falls to a non-operative level before the +5 volt line has dropped significantly. Resistor R2 and diode D1 serve as a low impedance discharge path for C1.

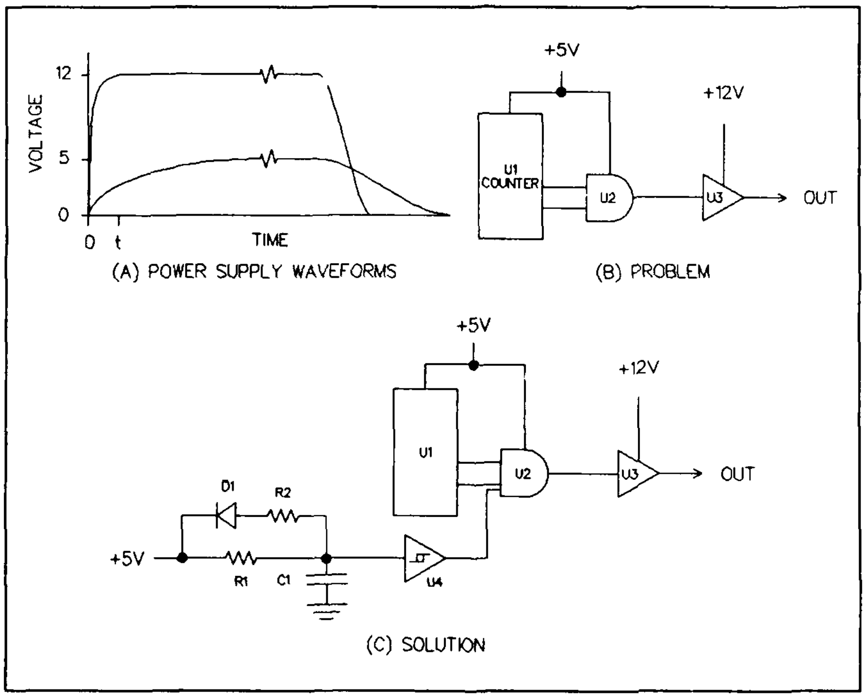


Figure B-17. INTERFACING CIRCUITS POWERED BY DIFFERENT SUPPLIES

SNEAK LABELS AND INDICATIONS

TARGET: Signal labels.

PROBLEM: Interface signals routed to unintended places.

SOLUTION: Check signal names on both sides of an interface.

COMMENT: Unintended routing is a form of sneak path typically involving apparent reversal of polarity or phase between signals crossing a subsystem interface. For example, assume signals labeled TRIG+ and TRIG- appear in the output of subsystem A. A problem occurs if the input required for subsystem B is labeled TRIG IN. Instead, the label should clearly indicate the desired polarity (e.g. TRIG+ IN or TRIG- IN) to avoid errors during analysis, assembly or maintenance actions.

TARGET: Indicators and associated drive circuitry.

PROBLEM: An indicator that monitors the commanded state of a function rather than the actual state.

SOLUTION: Insure that the indicator is monitoring the state of commanded function rather than the command signal.

COMMENT: For example, an indicator monitoring the relay coil circuit in Figure B-18(A) will only show the <u>intended</u> state of the relay. Instead, the indicator circuit should monitor the relay contact circuit as in Figure B-18(B) to show the <u>actual</u> state of the relay.

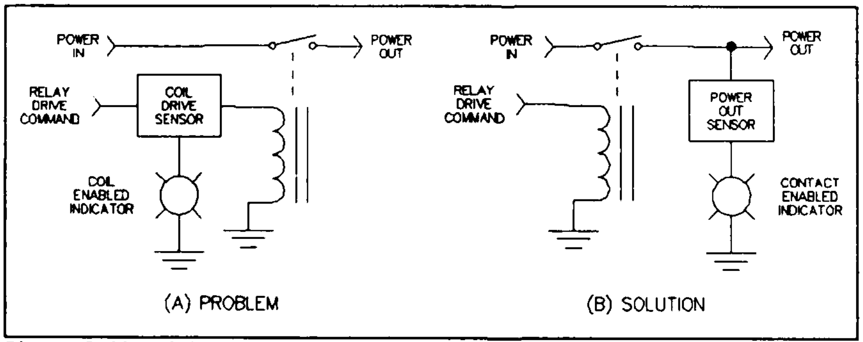


Figure B-18. INDICATORS

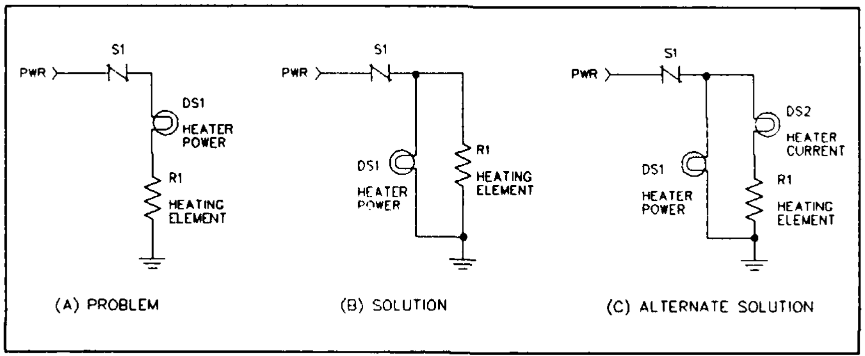
SNEAK LABELS AND INDICATIONS (Continued)

TARGET: Indicators and associated drive circuitry.

PROBLEM: An indicator circuit that depends upon the function it monitors for proper operation.

SOLUTION: Insure indicator power and drive signals are present even when the monitored function has been turned off, disconnected, or is inoperative due to a failure.

COMMENT: If an indicator circuit depends upon the operation of the monitored function, improper or unexpected operation of the function may inhibit the indicator circuitry. An example is shown in Figure B-19. In part (A) of the figure, assume heating element R1 has failed open. In this case, Heater Power lamp DS1 will indicate power off when in fact power is still available at the supply side of the heating element. This misleading indication presents a safety hazard to service personnel. A solution to this problem is shown in part (B) of the figure. To indicate heater failure without implying power off, the lamp circuit in part (A) should be used with DS1 labeled HEATER CURRENT. Alternatively, separate operations and maintenance indicators can be installed as shown in part (C).



(A) PROBLEM (B) SOLUTION (C) ALTERNATE SOLUTION

Figure B-19. FALSE INDICATION

B.3 SNEAK CIRCUIT DEVICE GUIDELINES

The Sneak Circuit Functional Guidelines aid the circuit designer or reliability analyst to identify devices commonly associated with sneak conditions. Each guideline is formatted as follows:

a. TARGET -- The types of circuit devices targeted by the guideline.

b. PROBLEM -- A statement of the sneak problem addressed by the rule.

c. SOLUTION -- A recommended approach for implementing the rule in practical situations.

d. COMMENT -- Supplementary information further explaining the rule, in some cases accompanied by a figure.

The following is a selection guide for application of the device guidelines:

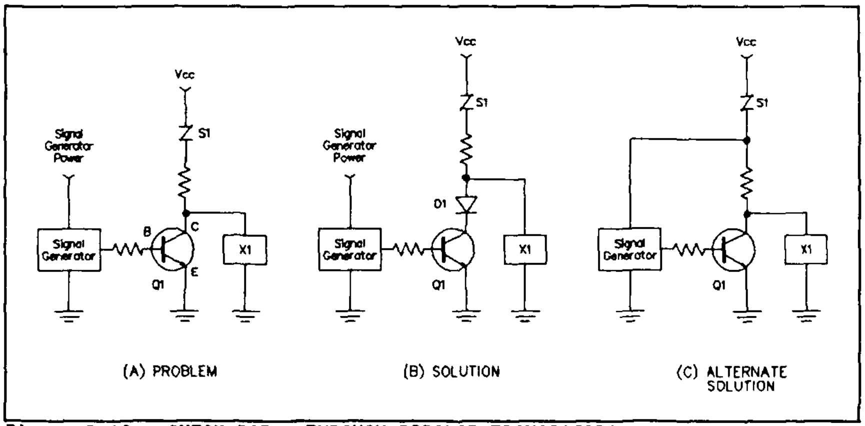
APPLICABLE DEVICES	
Analog Semiconductor Devices	B-26
Relay Timing	B-29
Digital Devices	B-30

TARGET: Bipolar NPN or PNP transistors.

PROBLEM: Sneak path in the forward direction through a normally reversed biased base-collector junction.

SOLUTION: Analyze the circuit to determine if the base-collector junction can become forward biased. If so, use a diode to prevent current flow out of the collector, or redesign the circuit to avoid the forward bias condition.

COMMENT: This problem can occur if collector voltage V_{CC} is removed from the collector and a signal is applied at the transistor base. In part (A) of Figure B-20, if collector bias voltage is removed from NPN transistor Q1 by opening switch S1, then signal generator current can flow through the base-collector junction of Q1 and into load X1. This current flow can be prevented as shown in part (B) of the figure by the addition of diode D1 or alternatively as shown in part (C) by powering the signal generator from the switched supply.



(A) PROBLEM

(B) SOLUTION

(C) ALTERNATE SOLUTION

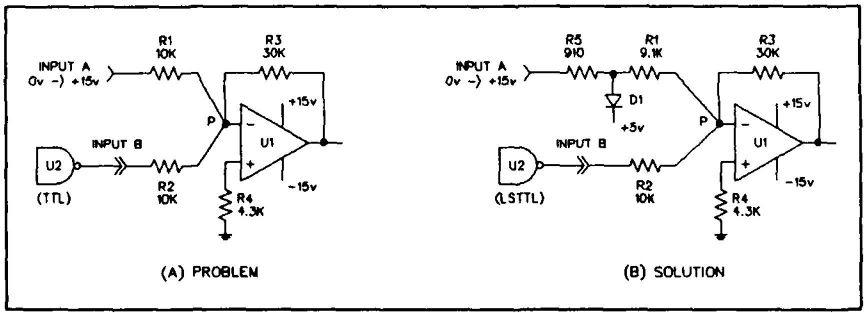
Figure B-20. SNEAK PATHS THROUGH BIPOLAR TRANSISTORS

TARGET: Operational amplifiers.

PROBLEM: **Presence of reverse current at the input summing point of an op-amp adder unintentionally driven into saturation.**

SOLUTION: **Limit the maximum amplitude of the input signal to the peak output swing of the amplifier divided by the closed loop gain or insure each input is capable of withstanding the highest voltage input source.**

COMMENT: The summing point of an op-amp adder (node P in Figure B-21) will remain at virtual ground so long as the op-amp is not driven into saturation. In part (A) of the figure, saturation will occur when input A exceeds 5v since the closed loop gain (-R3/R1 = -3) times the input voltage will exceed the op-amp negative supply (-15v). In this case current will flow from input A to input B, possibly damaging TTL gate U2. In part (B) of the figure, diode D1 clamps the signal to approximately 5.7 volts (5v plus one forward diode drop). Resistor R5 has been added to limit the current flowing through D1 when input A causes the diode to clamp. In addition, an LSTTL gate is substituted for the TTL version; the former can withstand up to 10 volts applied to its output while in a high state and can typically sink up to 8 milliamperes while in the low state.



(A) PROBLEM

(B) SOLUTION

Figure B-21. REVERSE CURRENT AT AN OP-AMP SUMMING POINT

ANALOG SEMICONDUCTOR DEVICES (Continued)

TARGET: Noise susceptible devices such as low level, high gain signal amplifiers, multivibrators (one-shots), thyristors (SCRs, triacs).

PROBLEM: Sensitive signal paths in close proximity to switched signal or power lines.

SOLUTION: Route sensitive paths away from noise sources. Shield sensitive signals using ground planes or guard bands on PC boards or cable shields for discrete wiring.

COMMENT: Noise can be capacitively or inductively coupled into a susceptible input from adjacent power lines or switched signal lines. Identify sneak electromagnetic paths by examining the physical layout of the circuitry. A schematic points to this problem if it depicts high current or high voltage devices on the same circuit board with susceptible devices. Examples of noise sources include:

a. power lines
b. brush-type motors
c. lamp, squib, stepper motor or other type of high current driver
d. Outputs of a digital counter (particularly when all bits change simultaneously)

Circuit elements that are particularly susceptible include:

a. op-amps
b. comparators (when an input is near the trip threshold)
b. one-shots
c. SCRs and triacs (gate input)
d. MOS type devices
e. clock lines

RELAY TIMING

TARGET: Relays, solenoids, contactors, stepper motors and other inductively actuated devices.

PROBLEM: **Relay coils with single or double diode suppression networks have long current decay time constants which can cause timing problems.**

SOLUTION: **Place a zener diode in series with the standard diode, cathode to cathode with the standard diode's anode connected to the negative end of the coil.**

COMMENT: When a relay is de-energized, a transient is induced in the coil. Because the transient could reduce the reliability of associated circuit components, transient suppression is normally required. A typical suppression technique shown in part (A) of Figure B-22 is to add a diode (D1 in the figure) across the coil (cathode connected to the positive side). An optional second diode (D2 in part (B) of the figure) can be added between the coil and power to protect the first diode against burn-out due to accidental reverse application of power. Adding a zener diode (D3) in series with the standard diode across the relay coil provides good suppression while decreasing relay drop-out time. With no zener, the L/R time constant increases by a factor of 5 to 10, causing contact bounce and early wear-out.

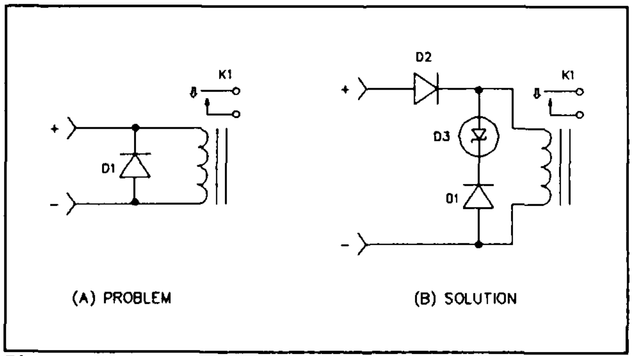


Figure B-22. RELAY SUPPRESSION NETWORKS

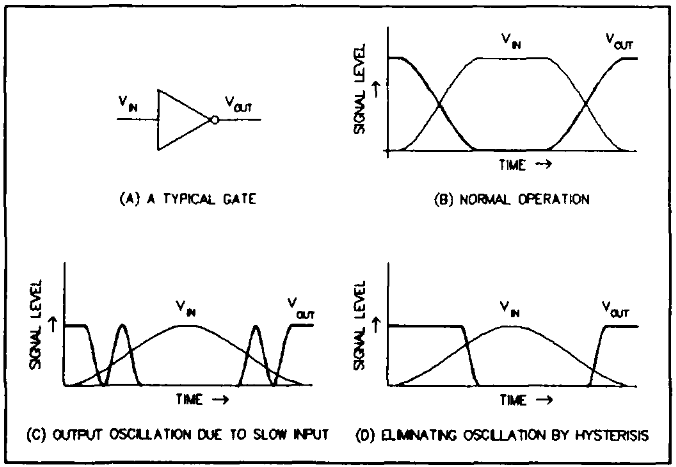
DIGITAL DEVICES

TARGET: TTL and MOS logic, memory and processor devices.

PROBLEM: Slow rise or fall times for any input signal to a digital device.

SOLUTION: Use Schmitt triggered gates for decreasing signal transition times. Avoid placing capacitors on logic signal paths.

COMMENT: Input signal rise or fall times slower than 50 nsec for TTL or 15 usec for MOS can cause multiple false triggering of the device or excessive power dissipation. This is depicted in Figure B-23. Part (A) of the figure shows a typical inverting gate with voltage V_{IN} applied to its input and V_{OUT} appearing at its output. The output waveform corresponding to an input signal having normal rise and fall times is shown in part (B). When the input rise and fall times are excessively slow as shown in part (C), an oscillation can occur at the output during the time interval that the input signal is between the unambiguous logic 0 and logic 1 states (i.e. for TTL, less than 0.8 volts and greater than 2 volts). As shown in part (D), this oscillation is avoided when a Schmitt triggered gate is used. This type of gate employs positive feedback to virtually eliminate input signal level ambiguity.



(A) A TYPICAL GATE

(B) NORMAL OPERATION

(C) OUTPUT OSCILLATION DUE TO SLOW INPUT

(D) ELIMINATING OSCILLATION BY HYSTERISIS

Figure B-23. EFFECT OF SLOW RISE OR FALL TIMES

PROBLEM: **Undesired triggering caused by open (floating) inputs.**

SOLUTION: **Terminate unused inputs to power or ground. Tie TTL inputs (except diode-input LSTTL) to power through a series resistance of from 1K to 5K ohms. Tie diode-input LSTTL devices directly to power or ground.**

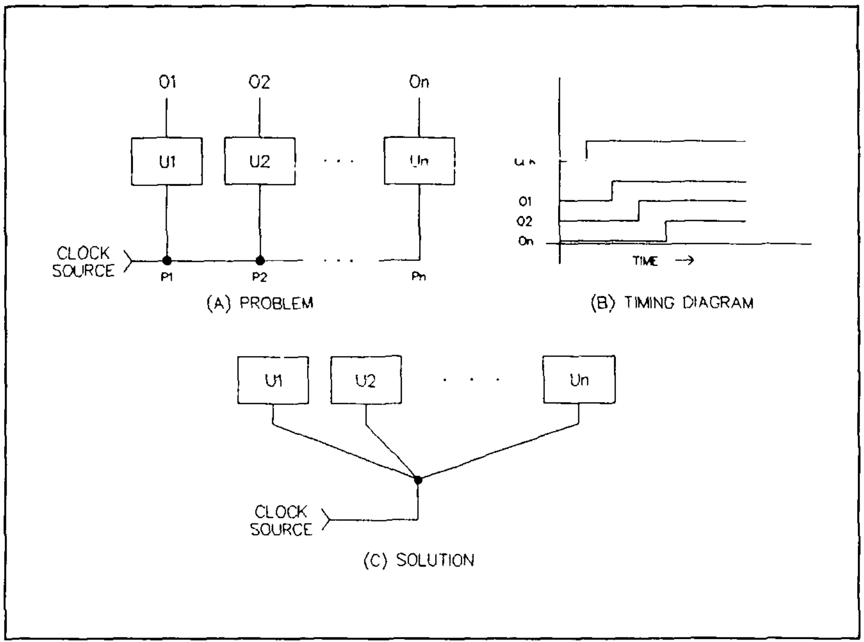
COMMENT: Floating inputs (particularly CMOS) can trigger a device because of charge coupling from nearby signals. Tying unused gate inputs with those that are used instead of to power or ground will increase input current and capacitive loading. This practice along with series resistance termination should be especially avoided for LSTTL devices with diode inputs because of their higher noise susceptibility (noise can enter through the parasitic capacitance associated with each diode).

PROBLEM: **Digital devices tied to a single, physically long clock signal path.**

SOLUTION: **Use separate, short, equal length runs from the clock source to each device.**

COMMENT: Propagation delay along a single clock path can cause timing skew between outputs of clocked devices. An example is shown in Figure B-24. In part (A) of the figure, the path length from the source of the clock signal to the clock input of U1 is shorter than for U2, which in turn is shorter than for U*n*. The resulting timing skew among outputs O1, O2 and O*n* is shown in part (B) of the figure. This skew can be reduced by equalizing the clock path lengths as shown in part (C). In general, path length equalization is necessary when the difference between the length of the signal path from the clock to the closest clocked device and the path from the clock to the farthest clocked device is greater than the following values:

Clock Frequency	Path Length
1 MHz	70 inches
10 MHz	7 inches
100 MHz	0.7 inches



(A) PROBLEM

(B) TIMING DIAGRAM

(C) SOLUTION

Figure B-24. ELIMINATING TIMING SKEW BY EQUALIZING CLOCK PATH LENGTHS

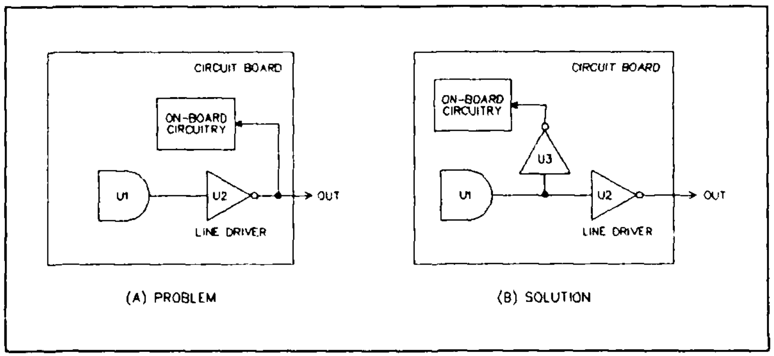
DIGITAL DEVICES (Continued)

TARGET: TTL and MOS logic, memory and processor devices.

PROBLEM: Erroneous signals generated by line drivers with outputs fed back to other on-board logic.

SOLUTION: Use line drivers only for interfacing with line receivers. Feed back signals from driver inputs, adding inverters if necessary. Do not use logic having internal feedback as a line driver.

COMMENT: Signal reflections on the transmission line can erroneously trigger logic either on a circuit board or on a driver chip that is tied to the driver output. Part (A) of Figure B-25 depicts on-board logic susceptible to false triggering. ln part (B) of the figure, the problem is avoided by driving the on-board circuitry with a signal derived from the driver U2 input. Inverter U3 is added both for obtaining the correct polarity ana for buffering the signal so as not to overload signal source U1.



(A) PROBLEM

(B) SOLUTION

Figure B-25. DIGITAL LINE DRIVERS

TARGET: TTL and MOS logic, memory and processor devices.

PROBLEM: Faulty operation or damage to a digital device due to high current flowing through the substrate diode at the device input.

SOLUTION: Insure inputs of digital devices are not driven below ground. Place fast recovery diodes at the device input (anode to input, cathode to one diode drop above ground) to prevent the input from going below ground.

COMMENT: The diodes supplement those that are on the chip; the latter are typically designed to handle short duration current transients while the former will protect against steady state currents. Negative going signals can occur on lines transmitting data from off the board or on lines coming from circuitry powered by a negative supply. The placement of the protection diode is shown in Figure B-26.

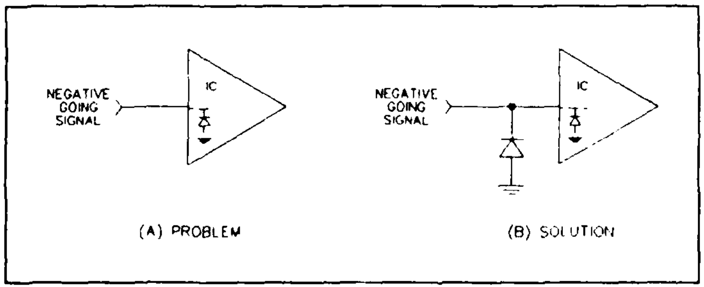


Figure B-26. IC INPUT CLAMPING DIODE

APPENDIX C

REFERENCES

BALD87 V. D. Baldwin, Long-Term Storage Effects on Fuzes, AFATL-TR-87-03 (AD-B110924L), 26 January 1987.

BOEI75 Boeing Co., Electrical Analysis of B-52/FB-111 AMAC and Release Circuitry Utilizing Sneak Circuit Analysis Techniques, Report D2-118576-1/2/3 (AD-A103951/2/3), October 1975.

BURA82 D. L. Buratti and S. G. Godoy, Sneak Analysis Application Guidelines, RADC-TR-82-179 (AD-A118479L), June 1982.

CLAR76 R. C. Clardy, "Sneak Circuit Analysis Development and Application," 1976 Region V IEEE Conference Digest, 1976, pp. 112-116.

CLAR80 R. C. Clardy, "Sneak Circuit Analysis," in J. E. Arsenault and J.A. Roberts (ed.), Reliability and Maintainability of Electronic Systems, Computer Science Press, 1980, pp. 223-241.

MILL89 J. Miller, Sneak Circuit Analysis for the Common Man Oct '89, RADC-TR-89-223

MILL90 J. Miller, Integration of Sneak Analysis with Design May '90, RADC-TR-90-109

MS1543B MIL-STD-1543B, Reliability Program Requirements for Space and Missile Systems, (Draft), 29 September 1987.

NP3634 Dept. of Navy, Sneak Circuit Analysis: A Means of Verifying Design Integrity, NAVSO P3634 (stock no. 0518-LP-394-8000), July 1986.

RANK70 J. P. Rankin and C. F. White, Sneak Circuit Analysis Handbook, Boeing Report D2-118341-1 (NTIS no. N71-12487), 15 July 1970.